

С. М. ВОЛЯНСЬКИЙ
Я. Б. ВОЛЯНСЬКА

МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ

НАВЧАЛЬНИЙ ПОСІБНИК

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування
імені адмірала Макарова

С. М. Волянський, Я. Б. Волянська

МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ

*Навчальний посібник
за редакцією доктора технічних наук,
професора О. А. Онищенка*

Миколаїв
«Іліон»
2018

УДК 62-83:629.584
В 72

РЕЦЕНЗЕНТИ:

- РОЖКОВ С. О., д-р. техн. наук, професор, завідувач кафедри експлуатації суднового електрообладнання та засобів автоматизації Херсонської державної морської академії;
- ЯРОВЕНКО В. О. д-р техн. наук, доцент, професор кафедри електротехніки та електрообладнання суден Одеського національного морського університету;
- ХЛОПЕНКО М. Я., д-р. техн. наук, професор, професор кафедри автоматизації Національного університету кораблебудування імені адмірала Макарова

*Рекомендовано Вченою радою НУК ім. адмірала Макарова
(протокол №10 від «27» жовтня 2017 р.)*

Волянський С. М.

В 72 Моделювання електромеханічних систем : навчальний посібник / С. М. Волянський, Я. Б. Волянська; за ред. проф. О. А. Онищенка. — Миколаїв : Іліон, 2018. — 246 с.

ISBN 978-617-534-483-5

У посібнику викладено особливості моделювання електромеханічних пристроїв і систем за допомогою програмного комп'ютерного середовища MATLAB та його пакетів розширення. Розглянуто основні принципи функціонального і структурного моделювання в пакеті Simulink, докладно розглянуто питання моделювання електричних кіл, напівпровідникових перетворювачів і електромеханічних пристроїв у пакеті Sim Power Systems. Висвітлено особливості дослідження систем автоматичного керування за допомогою пакета Control System Toolbox.

Навчальний посібник містить систематизований виклад основних положень, що повністю відповідають змісту робочої програми з дисципліни «Моделювання електромеханічних систем» для бакалаврів спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка».

УДК 62-83:629.584

ISBN 978-617-534-483-5

© Волянський С. М.,
Волянська Я. Б., 2018

ЗМІСТ

ВСТУП 7

Розділ 1.

ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ ЕЛЕМЕНТІВ ТА СИСТЕМ У ПАКЕТІ SIMULINK

1.1. Основні прийоми роботи в Simulink	9
1.1.1. Створення моделі	9
1.1.2. Налаштування параметрів блоків	10
1.1.3. Налаштування параметрів моделювання	11
1.1.4. Одержання результатів моделювання	13
1.2. Елементи функціональних моделей	18
1.2.1. Джерела сигналів	18
1.2.2. Комутаційні пристрої	21
1.2.3. Нелінійні елементи	24
1.2.4. Блоки верифікації сигналу	28
1.3. Моделювання систем автоматичного керування	31
1.3.1. Моделі динамічних об'єктів у пакеті Simulink	32
1.3.2. Дослідження систем автоматичного керування в пакеті Control System Toolbox	38
1.4. Підсистеми	42
1.4.1. Прості підсистеми	42
1.4.2. Керовані підсистеми	44
1.4.3. Масковані підсистеми	56
Контрольні питання до розділу 1	58
Завдання для самостійного виконання	59

Розділ 2.

МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ КІЛ ТА ЇХ ЕЛЕМЕНТІВ

2.1. Особливості застосування пакетів Sim Power Systems і Simulink для моделювання електротехнічних пристроїв	60
2.2. Моделювання електричних кіл однофазного синусоїдного струму	62

2.2.1. Моделі джерел синусоїдних електричних сигналів.....	63
2.2.2. Моделі резистивно-індуктивно-ємнісних кіл.....	64
2.2.3. Моделі вимикача.....	65
2.2.4. Моделі трансформатора.....	66
2.2.5. Моделі вимірювальних приладів.....	72
2.2.6. Одержання характеристик електричного кола.....	76
2.3. Моделювання трифазних електричних кіл.....	76
2.3.1. Моделювання трифазних джерел напруги.....	76
2.3.2. Моделювання елементів трифазного кола.....	81
2.3.3. Моделювання кіл у режимі короткого замикання.....	82
2.3.4. Моделювання трифазного вимірника струмів і напруг.....	84
2.4. Моделювання електричних кіл постійного струму в перехідному режимі.....	86
2.4.1. Установка початкових умов.....	87
2.4.2. Структурна модель кола.....	89
2.5. Моделювання несинусоїдних електричних кіл.....	91
2.5.1. Моделювання джерел несинусоїдних сигналів.....	91
2.5.2. Аналіз несинусоїдного кола.....	92
2.6. Моделювання нелінійних електричних кіл.....	94
2.6.1. Модель елемента, заданого в табличній формі.....	94
2.6.2. Модель елемента, заданого в аналітичній формі.....	96
2.7. Моделювання нестационарних електричних кіл.....	98
2.8. Сумісність елементів електричних кіл.....	100
Контрольні питання до розділу 2.....	101
Завдання для самостійного виконання.....	102

Розділ 3. МОДЕЛЮВАННЯ НАПІВПРОВІДНИКОВИХ ПЕРЕТВОРЮВАЧІВ

3.1. Елементи силової електроніки.....	103
3.1.1. Стандартні моделі елементів бібліотеки Power Electronics.....	103
3.1.2. Модель біполярного транзистора.....	109
3.2. Моделювання систем керування перетворювачами.....	111
3.2.1. Системи керування на основі Simulink-моделей.....	112
3.2.2. Системи керування на основі генератора імпульсів із синхронізацією від мережі Synchronized 6-pulse generator.....	120
3.2.3. Системи керування на основі ШІМ-генератора PWM Generator.....	125
Контрольні питання до розділу 3.....	131
Завдання для самостійного виконання.....	131

Розділ 4. ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ ПРИ МОДЕЛЮВАННІ Й ОПТИМІЗАЦІЇ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ

4.1. Завдання теоретичних досліджень.....	132
4.2. Математичні методи.....	136
4.3. Аналітичні методи.....	147
4.4. Методи оптимізації.....	154
4.4.1. Аналітичні методи оптимізації.....	154
4.4.2. Числові методи оптимізації.....	158
Контрольні питання до розділу 4.....	164

Розділ 5. МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ ТЕХНІЧНИХ ОБ'ЄКТІВ, ПРИСТРОЇВ І СИСТЕМ

5.1. Моделювання як метод наукового пізнання.....	166
5.2. Особливості застосування методу математичного моделювання.....	169
5.3. Особливості спостережень і вимірів при розробці моделі.....	170
5.4. Випадковість і невизначеність.....	171
5.5. Адекватність моделі.....	173
5.6. Загальна класифікація математичних моделей.....	174
5.7. Основні етапи математичного моделювання.....	177
5.7.1. Постановка проблеми та її якісний аналіз.....	177
5.7.2. Побудова математичної моделі.....	178
5.7.3. Математичний аналіз моделі.....	179
5.7.4. Підготовка вихідної інформації.....	179
5.7.5. Числове розв'язання.....	180
5.7.6. Аналіз числових результатів та їх застосування.....	180
5.8. Взаємозв'язок етапів моделювання.....	181
5.9. Роль математичного моделювання у прикладних дослідженнях.....	182
Контрольні питання до розділу 5.....	184

Розділ 6. МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ ТА ЇХ ЕЛЕМЕНТІВ

6.1. Моделювання електричних машин.....	186
---	-----

6.1.1. Загальні властивості стандартних моделей бібліотеки Machines	186
6.1.2. Машини постійного струму	188
6.1.3. Асинхронні машини	192
6.1.4. Синхронні машини	195
6.2. Моделювання первинних двигунів	200
6.2.1. Парова турбіна з регулятором	200
6.2.2. Дизель	203
6.2.3. Ініціалізація моделей електричних машин	204
6.3. Моделювання систем автоматизованого електропривода	209
6.3.1. Загальні властивості стандартних моделей електроприводів	209
6.3.2. Електропривод постійного струму	213
6.3.3. Електропривод змінного струму	218
Контрольні питання до розділу 6	222
Завдання для самостійного виконання	223

Розділ 7.

ЗАГАЛЬНІ ПРИНЦИПИ ІДЕНТИФІКАЦІЇ ЕЛЕКТРОТЕХНІЧНИХ ОБ'ЄКТІВ І СИСТЕМ

7.1. Узагальнений алгоритм ідентифікації	224
7.2. Терміни, визначення і завдання технічної ідентифікації	228
7.3. Методи вирішення завдань ідентифікації параметрів математичних моделей	234
7.4. Узагальнена структурна схема процесу ідентифікації	235
Контрольні питання до розділу 7	238
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	239
ДОДАТКИ	240

ВСТУП

Останнім часом у зв'язку із суттєвим розвитком інформаційних технологій усе більша увага приділяється використанню у вищій освіті засобів обчислювальної техніки. Комп'ютерні технології, в основі яких лежать прикладні програмні пакети, надають можливість більш глибокого вивчення технічних дисциплін. У зв'язку з цим курс «Моделювання електромеханічних систем» (MEMC) дає змогу студентам одержати навички автоматизованого проектування і дослідження систем за допомогою сучасного об'єктно-орієнтованого програмного забезпечення.

Як основний інструмент математичного моделювання в курсі MEMC пропонується програмне середовище MATLAB і його пакети розширення Simulink, Sim Power Systems, Control System Toolbox. Матеріал даного посібника орієнтований на використання версій MATLAB 2008b або 2010b. Проте матеріал підготовлений так, що буде корисний і при роботі як з попередніми (починаючи з версії 6.5), так і з більш новими версіями програми.

Пропонований навчальний посібник містить 7 розділів.

У першому розділі розглянуті основні принципи функціонального моделювання в пакеті Simulink середовища MATLAB. Інші розділи присвячені розробці так званих «віртуальних» моделей у пакеті Sim Power Systems. При цьому моделюються різні електромеханічні системи та їх елементи: електричні кола (розділ 2), напівпровідникові перетворювачі і системи керування ними (розділ 3), електричні машини і системи електропривода (розділ 6). Викладання матеріалу супроводжується практичними прикладами з ілюстраціями й описом параметрів налаштування, що дає змогу зробити матеріал зрозумілим і доступним для самостійного опанування й використання. Більшість моделей, що наводяться в посібнику як приклади, розроблені авторами особисто.

Крім того, навчальний посібник містить розділи, в яких розглядаються теоретичні дослідження при моделюванні та оптимізації

електромеханічних систем (розділ 4), методологія моделювання технічних об'єктів, пристроїв і систем (розділ 5) та загальні принципи ідентифікації електротехнічних об'єктів і систем (розділ 7).

Курс МЕМС базується на наступних дисциплінах: «Вища математика», «Теоретичні основи електротехніки», «Теорія електропривода», «Електроніка», «Теорія автоматичного керування». Матеріал даного посібника може використовуватися в курсовому і дипломному проектуванні, при вивченні практично всіх профільюючих дисциплін спеціальності. Посібник може бути корисний також аспірантам і викладачам.

Навчальний посібник з дисципліни «Моделювання електромеханічних систем» складено відповідно до програми дисципліни навчального плану підготовки фахівців зі спеціальності 141 «Електроенергетика, електротехніка та електромеханіка».

Розділ 1

ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ ЕЛЕМЕНТІВ ТА СИСТЕМ У ПАКЕТІ SIMULINK

1.1 Основні прийоми роботи в *Simulink*

Пакет *Simulink* є основою середовища MATLAB і призначений для математичного моделювання лінійних та нелінійних динамічних пристроїв, представлених структурною чи функціональною схемою. Назвемо такі моделі функціональними. Для моделювання в *Simulink* використовується набір візуальних об'єктів – бібліотека елементів, що складається з окремих блоків. Кожен блок бібліотеки характеризується своїм графічним зображенням (піктограмою), вікном настроювання параметрів і математичним описом. При цьому користувач позбавлений необхідності самостійно складати математичний опис системи, яка моделюється, оскільки в *Simulink* ця задача вирішується автоматично шляхом перенесення у графічне вікно стандартних блоків і з'єднання їх між собою відповідно до схеми досліджуваного пристрою чи системи. Таким чином, *Simulink* по суті являє собою візуально-орієнтовану мову програмування, яка завдяки широкому набору стандартних блоків і зручному графічному інтерфейсу, відкриває значні можливості з моделювання різноманітних технічних систем.

1.1.1 Створення моделі

Після запуску програмного середовища MATLAB відкривається його командне вікно *Command Window*. Для складання моделі насамперед треба завантажити браузер бібліотек *Simulink*. Це можна зробити двома способами:

- 1) скористатися спеціальною кнопкою на панелі інструментів вікна MATLAB;

- 2) ввести в командному рядку після значка «>>» команду *Simulink* і натиснути клавішу *Enter*.

Після цього відкривається вікно, у лівій частині якого знаходиться перелік розділів бібліотеки *Simulink*, а в правій частині відображається набір блоків, що входять до складу активного розділу бібліотеки. Кожен блок має власне вікно налаштування параметрів, що відкривається подвійним натисканням лівої клавіші миші на піктограмі блока. Нові параметри стають чинними після натискання кнопки *OK* у нижній частині вікна. Кнопка *Help* дає змогу відкрити відповідний розділ довідкової системи *MATLAB* з описом даного блока (англійською мовою).


У загальному випадку рекомендується наступна схема створення та налаштування параметрів моделей у *Simulink*:

- 1) описати на змістовному рівні логіку досліджуваної системи;
- 2) визначити перелік блоків, що відповідають її основним компонентам;
- 3) створити вікно нової моделі (меню *File* → *New* → *Model*);
- 4) за допомогою миші помістити у вікно моделі необхідні блоки і з'єднати їх між собою відповідно з функціональними зв'язками;
- 5) виконати налаштування й узгодження параметрів блоків (п. 1.1.2);
- 6) встановити необхідні значення параметрів моделювання (п. 1.1.3);
- 7) запустити процес моделювання кнопкою ► (*Start simulation*) на панелі інструментів вікна моделі.

1.1.2 Налаштування параметрів блоків

Параметри можна вводити у вікнах налаштування блоків безпосередньо в числовому вигляді або у вигляді змінних, значення яких задані в окремому файлі з розширенням *.m* – так званий *m*-файл. Доцільно використовувати *m*-файл для тих моделей, параметри яких потрібно неодноразово змінювати в процесі налагодження. Наприклад, щоб підібрати оптимальні коефіцієнти регуляторів, потрібно декілька разів запустити розрахунок моделі з різними значеннями коефіцієнтів. Очевидно, що всі зміни зручніше виконувати в одному вікні *m*-файлу, замість того, щоб кожного разу відкривати окремі вікна налаштування блоків моделі.

Новий *m*-файл створюється за допомогою меню *File* → *New* → *Model* вікна *MATLAB*. При цьому відкривається вікно редактора, в якому вводяться значення змінних, функції, арифметичні вирази, команди, текстові коментарі та ін. Приклад *m*-файлу наведений у п. 1.3.2 на рис. 1.29. Після знака «%» записуються коментарі, тобто невиконувани вирази, які у редакторі *m*-файлів автоматично виділяються зеленим кольором. Знак «;» запобігає відображенню результатів розрахунку *m*-файлу в командному вікні *MATLAB*. Усі імена змінних у *m*-файлі повинні бути ідентичними змінним, що введені у вікно налаштування блоків моделі.

Далі *m*-файл треба зберегти під довільним іменем і запустити на виконання командою меню *Debug* → *Run* або кнопкою  на панелі інструментів редактора. У результаті змінні, що задані у файлі, будуть занесені в робочу область *MATLAB* і можуть бути використані моделлю під час поточної робочої сесії. Таким чином, запускати модель на виконання можна тільки після того, як був запущений *m*-файл, що містить значення її параметрів.

Існує можливість реалізувати автоматичний запуск *m*-файлу при запуску моделі. Для цього необхідно у вікні моделі виконати команду меню *File* → *Model Properties*, у вікні, що з'явилося, відкрити вкладку *Callbacks* і в полі *Model initialization function* (або *InitFcn*) записати ім'я *m*-файлу (без розширення *.m*). У результаті кожного разу при запуску моделі на виконання попередньо буде виконуватися зазначений *m*-файл. При цьому необхідно, щоб він знаходився в поточній папці *MATLAB*, тобто в тій папці, яка відкривається при виконанні команди *Open* (за замовчуванням це папка *Work*). Змінити поточну папку можна за допомогою браузера *Current directory* в головному вікні *MATLAB*.

1.1.3 Налаштування параметрів моделювання

Параметри моделювання встановлюються в спеціальному вікні, що відкривається за допомогою меню *Simulation* → *Configuration Parameters* (у більш ранніх версіях програми – *Simulation Parameters*) вікна моделі. Вікно параметрів має декілька розділів. Розглянемо найбільш важливі компоненти розділу *Solver* (рис. 1.1).

Start time, *Stop time* – час початку і закінчення розрахунку моделі відповідно. Необхідно пам'ятати, що час моделювання – це умовна

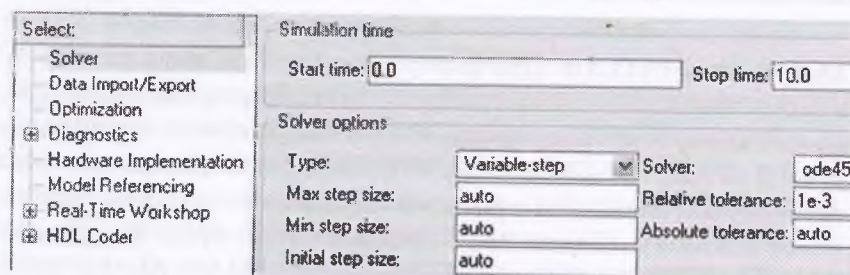


Рисунок 1.1 – Вікно налаштування параметрів моделювання (розділ *Solver*)

величина, він не відповідає реальному фізичному часові. Реальний час залежить від складності моделі, обраного методу і швидкодії комп'ютера, на якому виконується моделювання. В розглядуваній версії програми час закінчення моделювання встановлюється у спеціальному віконці на панелі інструментів вікна моделі, тому немає необхідності відкривати вікно параметрів.

Type – вид розв'язання. Можливі два варіанти:

- 1) *Variable-step* – розрахунок зі змінним кроком (для моделювання неперервних систем). Крок автоматично зменшується, якщо швидкість зміни результатів у процесі розрахунку зростає. І навпаки, якщо результати змінюються повільно, крок збільшується;
- 2) *Fixed-step* – розрахунок з фіксованим кроком (для дискретних систем).

Solver – метод моделювання. Якщо моделюється дискретна система, то у випадковому списку треба вибрати метод *discrete*. Методи *ode* застосовуються для розв'язування диференціальних рівнянь при розрахунку неперервних систем (*ode* – ordinary differential equations). За замовчуванням установлений метод Дорманда-Принса – *ode45*. Докладний опис усіх доступних методів і рекомендації з їхнього вибору наводяться в довідковій системі програми. У більшості випадків при моделюванні неперервних електромеханічних систем найкращі результати і максимальна швидкодія досягаються при виборі методу *ode23tb*.

Max step size, *Min step size*, *Initial step size* – максимальне, мінімальне і початкове значення кроку інтегрування відповідно. За замовчуванням дані параметри встановлені в *auto*. При цьому

програма вибирає значення кроків автоматично. Максимальне значення кроку розраховується за формулою

$$\Delta t_{\max} = (t_{\text{поч}} - t_{\text{кін}})/50,$$

де $t_{\text{поч}}$, $t_{\text{кін}}$ – початковий і кінцевий час моделювання, задані параметрами *Start time* і *Stop time* відповідно. Очевидно, що якщо моделюється періодичний процес і заданий час $t_{\text{кін}}$ набагато більший за період, то значення максимального кроку Δt_{\max} , обраного програмою автоматично, може виявитися занадто великим і навіть перевищувати значення періоду. Це може призвести до суттєвої втрати інформації про поведінку системи, і результат моделювання виявиться невірним. У таких випадках значення максимального кроку *Max step size* необхідно задавати самостійно, наприклад, у 10 разів менше від тривалості періоду. Однак варто враховувати, що занадто мале значення кроку може призвести до значного збільшення часу розрахунку моделі.

Relative tolerance, *Absolute tolerance* – відносна й абсолютна похибки інтегрування. На кожному інтервалі розрахунку програма визначає величину похибки для всіх змінних і порівнює його з допустимим значенням похибки, встановленим у даних полях налаштування. Якщо похибка більша за допустиме значення, то програма зменшує крок розрахунку й обчислює поточну точку заново. Параметр *Relative tolerance* подає оцінку похибки у відсотках щодо поточного значення змінної. За замовчуванням значення цього параметра задано 0,001. Це означає, що похибка не перевищує 0,1%. Параметр *Absolute tolerance* визначає абсолютну величину похибки, тобто різницю між розрахунковим й істинним значеннями змінної.

1.1.4 Одержання результатів моделювання

У результаті виконання сеансу моделювання користувач має змогу одержати інформацію про всі розраховані сигнали в даній моделі. Для цього призначені блоки-реєстратори, розташовані в бібліотеці *Sinks* пакета *Simulink*. Вони є універсальними, оскільки можуть фіксувати значення сигналів будь-якої фізичної природи. Існують кілька способів реєстрації розрахованого сигналу:

- 1) побудова графічної залежності;
- 2) одержання результату в числовому вигляді;
- 3) запис результату в робочу область MATLAB.

Основні віртуальні реєструючі прилади бібліотеки *Sinks* показані на рис. 1.2: *Scope* – осцилограф, *Display* – цифровий індикатор, *To Workspace* – блок запису сигналу в робочу область, *Out* – блок вихідного порту, *XY Graph* – графопобудовник. Основні властивості цих блоків розглянуті далі.

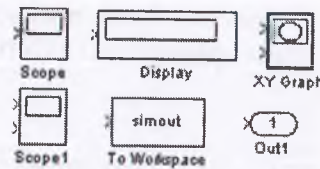


Рисунок 1.2 – Блоки бібліотеки *Sinks*

1.1.4.1 Подання результату в графічному вигляді

Найбільш корисний і часто використовуваний для цієї мети інструмент – віртуальний осцилограф *Scope* (рис. 1.3). Він подає результати моделювання у вигляді часових залежностей. Перевагою є можливість підключення осцилографа до будь-якої точки моделі, що спрощує процес налаштування. Подвійним натисненням миші на піктограмі блока відкривається координатна сітка, на якій після проведення сеансу моделювання відображається графік перехідного процесу. Автоматичне масштабування зображення виконується за допомогою кнопки (*Autoscale*) на панелі інструментів вікна осцилографа (див. рис. 1.3). Кнопки дають змогу вручну (за допомогою миші) змінювати масштаб або по двох осях одночасно, або по одній з осей – *X* чи *Y*. При цьому треба натиснути відповідну кнопку і виділити на графіку бажаний фрагмент.

Кнопка (*Parameters*) відкриває вікно налаштування параметрів. У поле *Number of axis* вводиться кількість необхідних вхідних портів, що дає змогу перетворити одноканальний осцилограф у багатоканальний (див. блок *Scope1* на рис. 1.2). При цьому кожному входу відповідає окрема координатна сітка.

На вкладці *Data history* вікна налаштування параметрів (рис. 1.4) встановлений прапорець *Limit data points to last*, що дає змогу

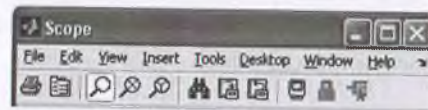


Рисунок 1.3 – Панель інструментів і меню блока *Scope*

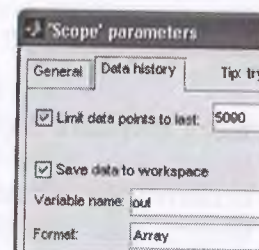


Рисунок 1.4 – Вікно налаштування параметрів блока *Scope*

вивести на екран задану кількість точок графіка (за замовчуванням 5000). Якщо кількість розрахованих значень перевищує указане число, то графік буде відображатися не повністю. Тому при виконанні складних моделей даний прапорець треба скинути.

Існує можливість шляхом додаткового налаштування блока *Scope* відкрити редактор властивостей *Property editor*, що дає змогу редагувати параметри осцилограм (колір графіків і фону, розмір шрифту, розбивку осей та ін.). Для цього треба, виділивши мишею в моделі блок *Scope*, у командному вікні MATLAB ввести команди

```
set(0,'ShowHiddenHandles','On')
set(gcf,'menubar','figure')
```

Далі натиснути клавішу *Enter*, потім відкрити даний блок. У його вікні над панеллю інструментів з'являється меню (див. рис. 1.3), в якому треба вибрати команди *Edit* → *Figure properties* і далі працювати з редактором властивостей, що дає змогу змінювати налаштування графіків. Для того, щоб скопіювати зображення в буфер і потім використовувати його в якій-небудь іншій програмі (наприклад, у *Microsoft Word*), треба вибрати пункт меню *Edit* → *Copy Figure*.

У версії MATLAB 2008b існує можливість зовсім обійтися без блока *Scope*, щоб не захаарувати вікно моделі. Це може бути корисно при роботі зі складними, багатокomпонентними моделями, коли потрібно відобразити велику кількість сигналів. При цьому використовується спеціальний інтерфейс *Signal & Scope Manager*. Відкрити його можна через меню *Tools* вікна моделі. Найбільш простий спосіб роботи з цим інтерфейсом – виділити в моделі необхідний сигнал і натиснути праву клавішу миші. Далі у контекстному меню вибрати послідовність команд *Create & Connect Viewer* → *Simulink* → *Scope*. При цьому відкривається вікно з координатною сіткою, у якій після проведення сеансу моделювання відображається перехідний процес. Щоб додати до тих же координатних осей інший сигнал, необхідно в його контекстному меню вибрати команду *Connect to existing viewer*.

Ще один пристрій для реєстрації процесів – це віртуальний графопобудовник *XY Graph* (див. рис. 1.2). На відміну від осцилографа, який будує залежність вхідного сигналу від часу $y = f(t)$, графопобудовник має входи по осях *X* і *Y*, що дає змогу прив'язувати значення двох вхідних сигналів до цих осей. У результаті будується

залежність $y = f(x)$, де $y = f(t)$, $x = f(t)$. Блок *XY Graph* може бути корисним при побудові фазових траєкторій, графіків у полярній системі координат, механічних характеристик електродвигунів у динаміці та ін. Координатна сітка даного блока відкривається автоматично при запуску процесу моделювання. Попередньо треба в його вікні налаштування задати мінімальні і максимальні значення по осях координат.

1.1.4.2 Подання результату в числовому вигляді

Для цієї мети призначений віртуальний індикатор *Display* – пристрій для реєстрації сигналу в числовому вигляді (див. рис. 1.2). Він зручний у тих випадках, коли сигнал змінюється дуже незначно протягом часу моделювання, тому немає необхідності в побудові графіка. Блок *Display* може використовуватися для виводу як скалярних, так і векторних даних. Якщо відображується величина є вектором (сукупністю сигналів), то в нижньому правому куті блока з'являється маленький чорний трикутник. У цьому випадку необхідно «розтягти» мишею зображення блока, щоб всі значення вектора стали видимі.

1.1.4.3 Запис результату в робочу область

Робоча область *Workspace* – це частина оперативної пам'яті комп'ютера, в якій зберігаються змінні упродовж поточної робочої сесії. З робочої області ці змінні можуть бути збережені у файлі, експортовані в яку-небудь іншу програму чи інший пакет розширення MATLAB, використані для керування зовнішніми пристроями в режимі реального часу тощо. За результатами моделювання, записаними у робочій області, можна також побудувати графік і відредагувати його за допомогою графічного редактора MATLAB, не використовуючи блок *Scope*. Змінні записуються в робочу область у формі таблиць, зміст яких можна переглянути в головному вікні MATLAB на вкладці *Workspace* (якщо ця вкладка відсутня, то треба установити прапорець в меню *Desktop* → *Workspace*).

При запуску процесу моделювання в робочій області автоматично формується змінна (вектор) з іменем *tout*, у якій містяться значення розрахункових моментів часу. За замовчуванням їхня кількість обмежена значенням 1000. Якщо за результатами моделювання

потрібно побудувати графік, то це обмеження необхідно зняти, оскільки вектор вихідної величини і вектор часу повинні мати однакові розмірності. Для цього треба відкрити вікно налаштування параметрів моделі (меню *Simulation* → *Configuration Parameters*) і в розділі *Data import/Export* зняти прапорець *Limit data points to last...* (у більш ранніх версіях програми даний розділ називався *Workspace I/O*).

Записати дані в робочу область можна декількома способами. Найпростіший – підключити до вихідного сигналу блок вихідного порту *Out1* (див. рис. 1.2) і запустити розрахунок моделі. Після виконання сеансу моделювання в робочій області, крім *tout*, з'являється змінна *yout*, що містить усі значення вихідного сигналу для моментів часу *tout*. Кількість стовпців матриці *yout* відповідає кількості сигналів, поданих на блок *Out1*. Після цього можна побудувати графік, записавши в командному вікні MATLAB команду

plot (tout, you).

Відкриється окреме вікно, в якому буде побудована залежність *yout* від *tout*. Для редагування графіка використовується кнопка *Edit plot* на панелі інструментів його вікна.

Другий спосіб занесення даних у робочу область – за допомогою блока *To Workspace* (див. рис. 1.2). У вікні налаштування цього блока необхідно задати ім'я змінної (за замовчуванням – *Simout*), а в полі *Save format* установити значення *Array*. Після сеансу моделювання змінна із заданим ім'ям з'явиться в робочій області. Для побудови графіка використовується команда

plot (tout, simout),

де замість *simout* записується ім'я змінної.

Третій спосіб – за допомогою осцилографа *Scope*. У вікні налаштування параметрів цього блока треба відкрити вкладку *Data history* (рис. 1.4), установити прапорець *Save data to workspace* і задати ім'я змінної (наприклад, *out*). Параметр *Format* установити в значення *Array*. Після розрахунку моделі задана в робочій області змінна *out* буде містити в собі два стовпці: у першому – значення часу, у другому – значення вихідного сигналу. Для побудови графіка стовпці треба розділити і зберегти в окремих векторах. Для цього в командному вікні ввести команди:

$t = out(:,1)$ – в матриці out виділяється перший стовпець (значення часу) і зберігається у вигляді вектора з іменем t ;

$y = out(:,2)$ – в матриці out виділяється другий стовпець (значення вихідного сигналу) і зберігається у вигляді вектора з іменем y ;

$plot(t, y)$ – побудова графіка $y = f(t)$.

Введення кожного рядка супроводжується натисненням клавіші *Enter*.

1.2 Елементи функціональних моделей

1.2.1 Джерела сигналів

Моделі джерел впливів знаходяться в бібліотеці *Sources* пакета *Simulink*. У залежності від специфіки розв'язуваної задачі, впливи на виході даних блоків можуть імітувати сигнали різної фізичної природи – напругу, температуру, тиск, переміщення та ін. Найбільш часто використовувані елементи бібліотеки *Sources* наведені в табл. 1.1.


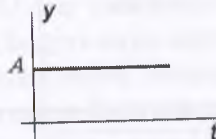

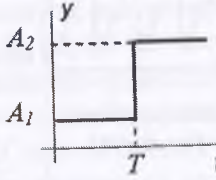

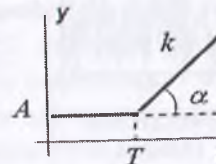

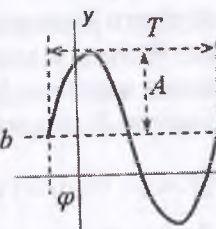

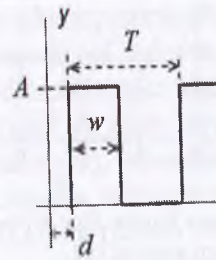
Розглянемо деякі особливості блоків, наведених у табл. 1.1.

Step – джерело одиничного перепаду; генерує сигнал, що у заданий момент часу стрибком змінює своє значення. Для задання від'ємного перепаду початкові значення можуть бути як додатними, так і від'ємними.


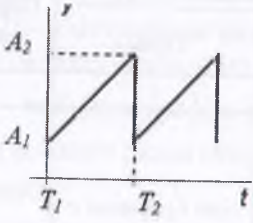

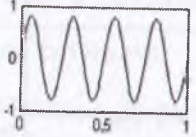
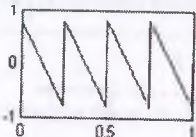
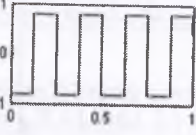
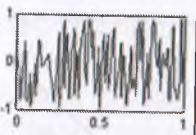
Ramp – джерело лінійно-наростаючого впливу; параметр *Slope* (нахил) є тангенсом кута нахилу графіка сигналу до осі абсцис. При від'ємному значенні даного параметра сигнал з часом буде зменшуватися.

Sine wave – джерело синусоїдного сигналу; у полі *Frequency* задається кутова частота коливань Ω , що зв'язана з періодом T і частотою f співвідношенням $\Omega = 2\pi / T = 2\pi f$. У версії MATLAB, що розглядається, у налаштуваннях значення цього сигналу (*Initial value*) має бути більше кінцевого (*Final value*). В обох блоках присутній додатковий параметр *Time* (рис. 1.5, а), який визначає джерело значень часу для розрахунку синусоїдної залежності: *Use simulation time* – використовувати час моделювання, *Use external signal* – використовувати зовнішній сигнал. В останньому випадку в блока з'являється вхід, на який подаються значення часу від зовнішнього джерела.

Таблиця 1.1 – Елементи бібліотеки *Sources* пакета *Simulink*

Зовнішній вигляд блока	Вихідний сигнал		Параметри вікна налаштування
	Залежність $y = f(t)$	Опис	
1	2	3	4
 Constant		$y = A$ при будь-яких t	Constant value – A
 Step		$y = A_1$ при $t < T$, $y = A_2$ при $t > T$	Step time – T Initial value – A_1 Final value – A_2
 Ramp		$y = A$ при $t \leq T$, $y = k \cdot t$ при $t > T$ $k = \tan \alpha$	Slope – k Start time – T Initial output – A
 Sine wave		Синусоїдний сигнал: $y = A \cdot \sin(\Omega \cdot t + \varphi) + b$	Amplitude – A Bias – b Frequency – Ω (rad/s) Phase – φ (rad)
 Pulse Generator		Прямокутний сигнал	Amplitude – A Period – T (sec) Pulse Width – w (% of period) Phase delay – d (sec)

· Продовження табл. 1.1

1	2	3	4
 Repeating Sequence		Пилкоподібний сигнал	Time values – $[T_1, T_2]$ Output values – $[A_1, A_2]$ $T_1 < T_2$
 Signal Generator	 Синусоїдний (sine)	 Пилкоподібний (sawtooth)	Wave form: – sine – square – sawtooth – random Amplitude; Frequency (units): – rad/sec – Hertz
	 Прямокутний (square)	 Випадковий (random)	

Pulse Generator – джерело прямокутного сигналу; ширина імпульсу визначається параметром *Pulse Width* у відсотках від тривалості періоду. За замовчуванням задане значення 50, тобто тривалість імпульсу дорівнює тривалості паузи. В даному блоці, як і в

Time (t) Use simulation time

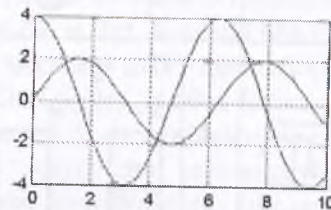
Amplitude:
[2.4]

Bias:
0

Frequency (rad/sec):
1

Phase (rad):
[0, pi/2]

а)



б)

Рисунок 1.5 – Вікно налаштування блока *Sine Wave* (а) та осцилограми на його виході (б)

Sine wave, також присутній параметр *Time*. Імпульси однополярні, тобто вихідний сигнал приймає лише нульові чи додатні значення.

Repeating Sequence – джерело пилкоподібного сигналу; у вікні налаштування задаються значення часу і відповідні їм значення вихідного сигналу для першого імпульсу послідовності. Причому, якщо в рядку *Output values* перше значення менше другого ($A_1 < A_2$), тоді імпульси будуть мати наростаючий передній фронт (див. рис. у табл. 1.1), а якщо перше значення більше другого ($A_1 > A_2$), тоді задній фронт імпульсів буде спадаючим.

Signal Generator – універсальний генератор. Форма сигналу на виході залежить від налаштування параметра *Wave form* (див. табл. 1.1). Крім форми сигналу, задається його амплітуда, частота, а також одиниці виміру частоти – Гц або рад/с. Задання від'ємної амплітуди призводить до фазового зсуву на 180° . Прямокутний сигнал на виході даного блока є двополярним імпульсом (на відміну від блока *Pulse Generator*), пилкоподібний сигнал є імпульсом з лінійно-спадаючим заднім фронтом.

На виході кожного блока (див. табл. 1.1), крім *Repeating Sequence*, при необхідності можна формувати не один, а декілька сигналів. Для цього необхідно у вікні налаштування блока задати значення параметрів окремо для кожного сигналу, розділивши їх пробілом або комою, у квадратних дужках. Так, наприклад, для одержання на виході блока *Constant* трьох сигналів зі значеннями 1, 2, 3 треба в полі *Constant value* ввести запис [1 2 3]. Як приклад на рис. 1.5 наведене вікно налаштування блока *Sine wave* і осцилограми на виході, які відповідають заданим параметрам.

1.2.2 Комутаційні пристрої

У бібліотеці *Signal Routing*, як випливає з її назви, знаходяться блоки, що виконують маршрутизацію сигналу. Іншими словами, з їхньою допомогою можуть бути реалізовані різні варіанти обміну даними між об'єктами моделі. Наприклад, збереження значень в оперативній пам'яті комп'ютера і витягнення з неї (блоки *Data Store Memory/Read/Write*), «безпроводна» передача даних (блоки *Goto, From*), об'єднання сигналів у спільну шину або вибір визначеного сигналу із шини (блоки *Merge, Bus Selector/Creator, Index Vector, Mux, Demux* та ін.). У даному параграфі розглянемо кілька блоків

бібліотеки *Signal Routing*, що виконують функцію перемикачів сигналів, оскільки вони можуть бути найбільш корисними при реалізації різних логічних структур у системах керування електромеханічними об'єктами.

1.2.2.1 Керований перемикач Switch

Блок *Switch* (рис. 1.6) має три входи – два інформаційних (u_1 та u_3) і один керуючий (u_2), та один вихід. Вікно налаштування блока показане на рис. 1.7. Логіка його роботи полягає в наступному: якщо сигнал, який надходить на вхід u_2 , задовольняє умові, заданій параметром *Criteria for passing first input*, то на вихід *out* блока передається сигнал з 1-го входу (u_1), в іншому випадку – сигнал з 3-го входу (u_3). У полі *Threshold* задається граничне значення – як числова константа або математичний вираз. Як критерій для проходження на вихід сигналу з входу u_1 у полі *Criteria for passing first input* вибирається одна з наступних умов:

$u_2 > \text{Threshold}$ («більше або дорівнює»),

$u_2 > \text{Threshold}$ («більше»),

$u_2 \sim \text{Threshold}$ («не дорівнює»).

За замовчуванням задана перша умова, а параметр *Threshold* встановлений у 0 (див. рис. 1.7). Для цього випадку алгоритм роботи блока можна скорочено записати в такому вигляді:

якщо $u_2 \geq 0$, то $out = u_1$;

якщо $u_2 < 0$, то $out = u_3$.

Як приклад на рис. 1.6 наведені два блоки *Switch* з однаковими налаштуваннями параметрів, заданими за замовчуванням, і однаковими сигналами на інформаційних входах. Розрізняються тільки керуючі сигнали.

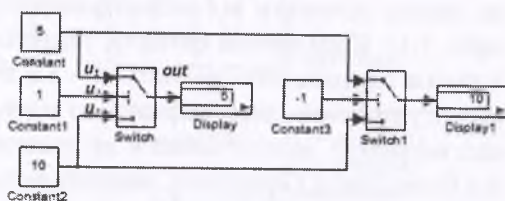


Рисунок 1.6 – Модель з використанням блока *Switch*

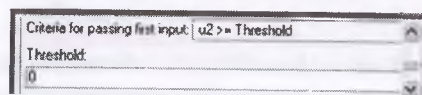


Рисунок 1.7 – Вікно налаштування блока *Switch*

Видно, що в залежності від значення u_2 на вихід ключа надходить сигнал або з входу u_1 , або з входу u_3 . При цьому положення перемикача на піктограмі блока не змінюється.

1.2.2.2 Багатовходовий перемикач Multiport Switch

Верхній вхід u_0 блока *Multiport Switch* – керуючий, інші – інформаційні (рис. 1.8). Кількість інформаційних входів n визначається параметром *Number of inputs*. Блок підключає до виходу той інформаційний вхід, номер якого дорівнює значенню сигналу, що надходить на керуючий вхід u_0 . Іншими словами,

$$out = u_i,$$

де значення $i = 1, 2, 3, \dots, n$ надходить на вхід u_0 . Очевидно, що значення керуючого сигналу не повинне перевищувати кількість входів ($u_0 \leq n$), інакше буде видане повідомлення про помилку.

У прикладі на рис. 1.8 на вхід u_0 блока *Multiport Switch* надходить значення 2, отже, до виходу підключається другий інформаційний вхід (u_2), тому вихідний сигнал дорівнює 15. Положення перемикача на піктограмі блока не змінюється в залежності від входу, що комутується.

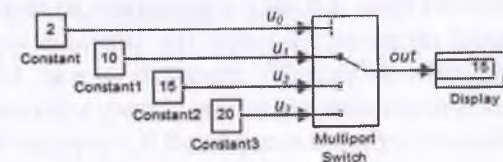
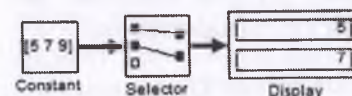


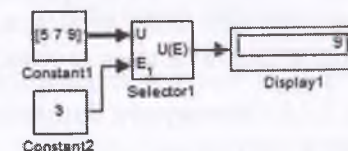
Рисунок 1.8 – Модель з використанням блока *Multiport Switch*

1.2.2.3 Селектор Selector

З виходу блока *Selector* (рис. 1.9) знімаються ті вхідні сигнали, номери яких задані в полі *Elements* його вікна налаштування в квадратних дужках (рис. 1.10).



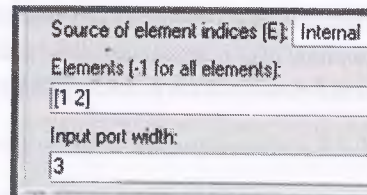
а)



б)

Рисунок 1.9 – Моделі з використанням блока *Selector*

При цьому блок має тільки один інформаційний вхід, на який надходить векторний сигнал. Кількість окремих сигналів у вхідному векторі визначається параметром *Input port width*. Якщо потрібно передати на вихід всі вхідні сигнали, то в поле *Elements* вводиться значення -1 . Для моделі з рис. 1.9, а вікно налаштування блока *Selector* показано на рис. 1.10. Оскільки в полі *Elements* задані значення 1 і 2, то селектор вибирає з вхідного вектора перший та другий сигнали (які дорівнюють 5 і 7 відповідно) і пропускає їх на вихід. При цьому піктограма блоку відображає поточне положення перемичників.

Рисунок 1.10 – Вікно налаштування блока *Selector*

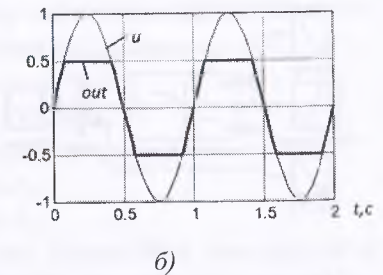
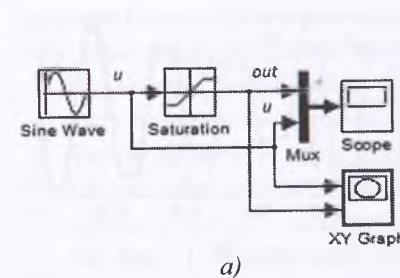
Можливе також зовнішнє задання номерів вхідних сигналів. Для цього потрібно встановити параметр *Source of element indices* у значення *External*. У такому випадку блок змінює свій зовнішній вигляд (рис. 1.9, б) і з'являється додатковий керуючий вхід E , на який подаються номери тих сигналів на вході U , які потрібно пропустити на вихід. У прикладі на рис. 1.9, б на керуючий (нижній) вхід надходить значення 3, тому з вхідного вектора даних блок вибирає третій сигнал, рівний 9, і передає його на вихід.

1.2.3 Нелінійні елементи

Пакет *Simulink* призначений переважно для моделювання нелінійних динамічних систем, тобто таких систем, у яких залежність між вихідним і вхідним сигналом має нелінійний характер. Більшість нелінійних блоків зібрані в бібліотеці *Discontinuities*. Деякі з них, наприклад *Relay* або *Quantizer*, характеризуються розривною передатною функцією. Усі нелінійні блоки даної бібліотеки прийняті ідеальними, тобто не враховується інерційність пристроїв, які представляються цими елементами. Розглянемо декілька найбільш розповсюджених блоків та приклади моделей з їх використанням.

1.2.3.1 Обмежувачі *Saturation* і *Saturation Dynamic*

Блок *Saturation* (рис. 1.11, а) являє собою модель нелінійного пристрою, який виконує обмеження величини вхідного сигналу.

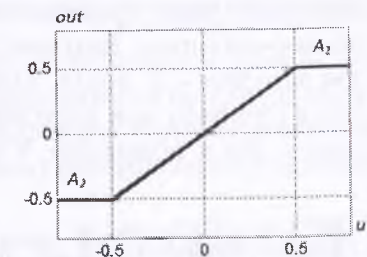
Рисунок 1.11 – Модель з використанням блока *Saturation* (а) і графіки перехідного процесу (б)

Сигнал *out* на виході блока дорівнює вхідному сигналу u до тих пір, поки вхідний не перевищує поріг обмеження.

Після досягнення вхідним сигналом рівня обмеження вихідний сигнал блока перестає змінюватися і залишається рівним граничному значенню. Верхній поріг обмеження A_1 задається параметром *Upper limit*, нижній поріг A_2 – параметром *Lower limit*. Алгоритм роботи блока можна записати в скороченому вигляді:

$$\begin{aligned} out &= u \text{ при } A_2 < u < A_1; \\ out &= A_1 \text{ при } u \geq A_1; \\ out &= A_2 \text{ при } u \leq A_2. \end{aligned}$$

На рис. 1.11, а показана модель, яка ілюструє роботу обмежувача, а на рис. 1.11, б – осцилограми в даній моделі. У налаштуваннях блока *Saturation* як пороги задані значення: $A_1 = 0.5$; $A_2 = -0.5$. Видно, що блок «обрізає» верхню частину синусоїдного сигналу, яка виходить за межі заданого діапазону $[-0.5 \dots 0.5]$. На рис. 1.12 наведена характеристика даного блока (залежність вихідного сигналу від вхідного $out = f(u)$), яка отримана за допомогою графо-побудовника *XY Graph*.

Рисунок 1.12 – Характеристика блока *Saturation*

Блок *Saturation Dynamic* (рис. 1.13, а) виконує аналогічну функцію, але границі обмеження не задаються у вікні налаштування, а надходять на верхній *up* і нижній *lo* входи блока у вигляді сигналів

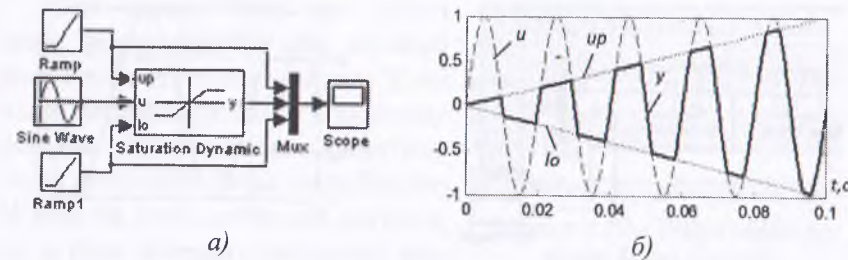


Рисунок 1.13 – Модель з використанням блока *Saturation Dynamic* (а) та графіки перехідного процесу (б)

від зовнішніх джерел. Даний блок може бути корисним, коли обмеження носить змінний характер або наперед є не відомим і має бути розрахованим у процесі моделювання. В моделі на рис. 1.13, а верхня і нижня границі задаються джерелами лінійно-наростаючого впливу *Ramp* і *Ramp1*, у яких швидкості наростання сигналів (параметр *slope*) встановлені в значення 10 і -10 відповідно. Осцилограми зображені на рис. 1.13, б.

1.2.3.2 Реле *Relay*

Блок *Relay* (рис. 1.14, а) реалізує релейну нелінійність.

Він працює за аналогією зі звичайним реле: якщо вхідний сигнал перевищує деяке граничне значення, то на виході формується дозволяючий сигнал. Вихідний сигнал *out* блока *Relay* може приймати два значення – A_1 або A_2 . Перехід з одного стану в інший відбувається стрибком при досягненні вхідним сигналом граничних значень x_1 або x_2 відповідно. У тому випадку, коли граничні значення

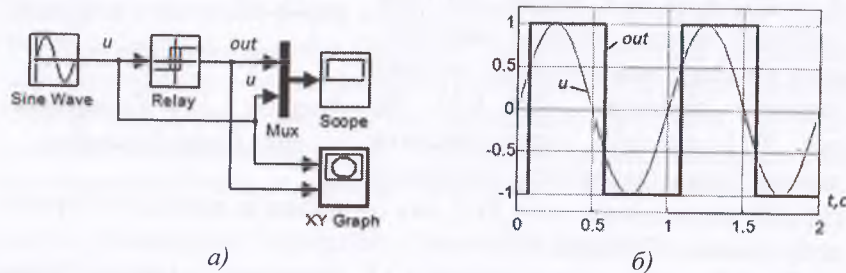


Рисунок 1.14 – Модель з використанням блока *Relay* (а) та графіки перехідного процесу (б)

не рівні ($x_1 \neq x_2$), блок реалізує релейну характеристику з гістерезисом. При цьому має виконуватися нерівність $x_1 > x_2$.

Параметри блока *Relay*:

Switch on point – поріг вхідного сигналу x_1 ;

Switch off point – поріг вхідного сигналу x_2 ;

Output when on – вихідний сигнал A_1 ;

Output when off – вихідний сигнал A_2 .

На рис. 1.14 показана модель, яка демонструє використання блока *Relay*, та осцилограми, на рис. 1.15, а – вікно налаштування блока в даній моделі, а на рис. 1.15, б – його характеристика отримана за допомогою блока *XY Graph*.

За замовчуванням у налаштуваннях блока *Relay* параметри x_1 і x_2 встановлені в значення ϵ_{ps} , що відповідає $2,22 \cdot 10^{-16}$, тобто практично дорівнює нулю.

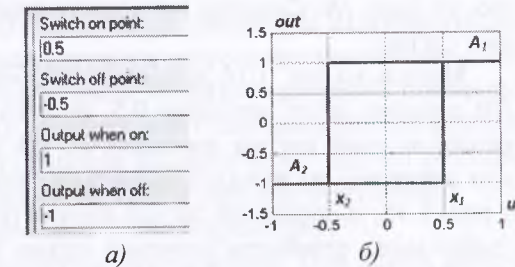


Рисунок 1.15 – Вікно налаштування (а) та характеристика (б) блока *Relay*

1.2.3.3 Індикатор перетинання граничного значення *Hit Crossing*

Блок *Hit Crossing* (рис. 1.16, а) фактично є блоком верифікації сигналів, більшість з яких знаходяться в бібліотеці *Model Verification* (див. п. 1.2.4). Однак розроблювачі *Simulink* помістили цей блок у бібліотеку *Discontinuities*, тому розглянемо його в даному розділі.

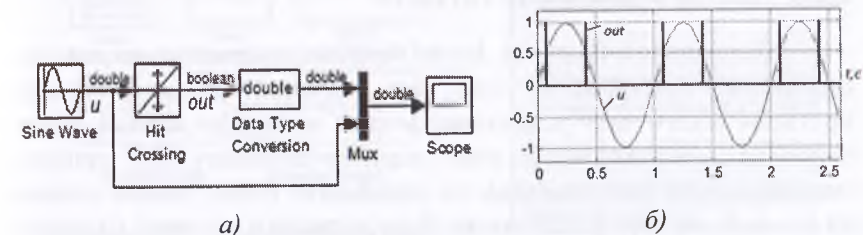


Рисунок 1.16 – Модель з використанням блока *Hit Crossing* (а) та графіки перехідного процесу (б)

Блок *Hit Crossing* визначає момент часу, коли вхідний сигнал перетинає граничне значення, задане в полі *Hit crossing offset*. Напрямок перетинання задається параметром *Hit crossing direction*:

rising – зростання сигналу;

falling – спадання сигналу;

either – обидва напрями.

У момент перетинання граничного рівня в заданому напрямку блок генерує одиничний сигнал тривалістю в один крок модельного часу. Якщо вхідний сигнал залишається рівним граничному значенню, то блок *Hit Crossing* продовжує генерувати на виході одиничний сигнал.

Модель на рис. 1.16 демонструє роботу даного блока. Параметр *Hit crossing offset* дорівнює 0,5, напрям перетинання – *either*. Як тільки вхідний сигнал дорівнює 0,5 (як при його наростанні, так і при спаданні), на виході з'являється одиничний імпульс. Як видно з рис. 1.16, а, вихідний сигнал блока *Hit Crossing* має тип *boolean* (тобто може приймати значення тільки 0 чи 1), а на виході блоку *Sine Wave* – тип сигналу *double*. При спробі об'єднати два сигнали різних типів в один вектор за допомогою мультиплексора *Mux* (з метою виводу їх на один екран осцилографа) виникає повідомлення про помилку. Ця проблема вирішується одним з двох способів. Перший спосіб – включити на виході блока *Hit Crossing* перетворювач сигналу *Data Type Conversion* (бібліотека *Signal Attributes*), у вікні налаштування якого встановити параметр *Output data type mode* у значення *double*. Другий спосіб – зняти прапорець *Implement logic signals as boolean data* у розділі *Optimization* вікна налаштування параметрів моделі (меню *Simulation* → *Configuration Parameters*).

1.2.4 Блоки верифікації сигналу

Елементи бібліотеки *Model Verification* призначені для підтвердження (верифікації) того, що значення вхідного сигналу протягом усього часу розрахунку моделі відповідає певній умові: наприклад, не виходить за межі заданого діапазону або, навпаки, знаходиться за його межами, не перевищує певне задане значення і т. п. Блок верифікації може бути корисним на етапі налаштування моделі, а потім, якщо модель функціонує належним чином, його можна дезактивувати. При цьому немає необхідності фізично

видаляти цей блок з моделі, достатньо встановити спеціальний прапорець у його вікні налаштування. При порушенні заданої умови (наприклад, вихід сигналу за встановлені межі) можливі два варіанти дій: зупинка моделювання з видачею відповідного повідомлення або генерація на виході блока нульового сигналу (при дотриманні умови сигнал дорівнює 1). В останньому випадку в блоці верифікації присутній спеціальний вихід для підтверджуючого сигналу. Це дає змогу застосовувати дані блоки при моделюванні різних систем керування, в яких сигнал підтвердження може бути використаний як керуючий імпульс.

Розглянемо блок *Check Static Range* (перевірка статичного діапазону, рис. 1.17), вікно налаштування якого зображене на рис. 1.18. У полях *Upper bound* і *Lower bound* вводяться значення верхньої і нижньої границь діапазону, у межах якого має знаходитися вхідний сигнал. Якщо встановлені прапорці *Inclusive upper bound* і *Inclusive lower bound*, то даний діапазон містить у собі обидві границі.

Далі розглянемо призначення інших параметрів.

Enable assertion – включити підтвердження. При знятті даного прапорця блок відключається, тобто ніяк не виявляє себе в процесі розрахунку моделі, хоча присутній у ній. При необхідності його можна знову включити.

Simulation callback when assertion fails – поле для введення команди або виразу мовою MATLAB, що буде виконуватися при порушенні умови (виходу сигналу за межі діапазону). Якщо ніяка дія не потрібна, то дане поле можна залишити порожнім.

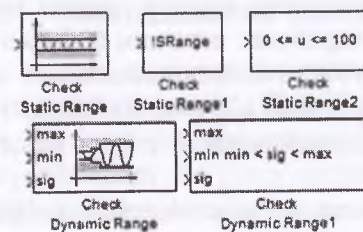


Рисунок 1.17 – Зовнішній вигляд блоків *Check Static Range* та *Check Dynamic Range*

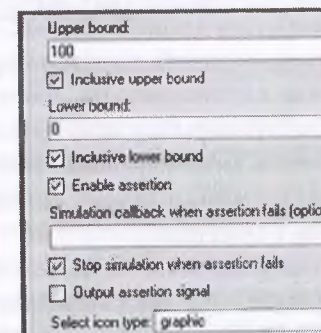


Рисунок 1.18 – Вікно налаштування блока *Check Static Range*

Stop simulation when assertion fails – зупинити моделювання при порушенні умови. При цьому, якщо прапорець встановлений, процес розрахунку моделі припиняється і видається повідомлення про помилку. Якщо прапорець знятий, то моделювання продовжується, а повідомлення виводиться в командному вікні.

Output assertion signal – вивести підтверджуючий сигнал. При установці прапорця в блоці з'являється вихідний порт, з якого знімається сигнал типу boolean: при виконанні умови він дорівнює 1, при порушенні – 0.

Select icon type – дозволяє вибрати вигляд піктограми блока: графічний або текстовий. У другому випадку на піктограмі відображається математичний запис підтверджуваної умови. Якщо текст не вміщається в розмір блока, то на його зображенні виводиться знак оклику. У цьому разі блок потрібно збільшити (див. рис. 1.17).

Інші блоки бібліотеки *Model Verification* мають схожі вікна налаштування і відрізняються один від одного умовами, яким має відповідати вхідний сигнал. Блок *Check Dynamic Range* (див. рис. 1.17) аналогічний розглянутому вище, з тією лише різницею, що верхня і нижня границі діапазону не задаються у вікні налаштування, а надходять у вигляді окремих сигналів на спеціальні входи *max* і *min*. Ці сигнали можуть змінюватися в часі, тому діапазон є динамічним. Існує також ряд блоків, у яких задається лише одна границя – верхня або нижня, причому вона може бути як статичною, так і динамічною. При цьому основний сигнал, що надходить на вхід *sig*, повинен задовольняти умові $sig < max$ (для блоків *Check Static /Dynamic Upper Bound*), або $sig > min$ (для блоків *Check Static /Dynamic Lower Bound*). Два інших блоки – *Check Static /Dynamic Gap* – також перевіряють заданий діапазон, однак на відміну від розглянутих вище блоків, підтверджують, що сигнал перебуває поза цим діапазоном. Тобто значення сигналу на кожному кроці моделювання мають знаходитися вище верхньої границі ($sig > max$) або нижче нижньої границі ($sig < min$). У цих випадках підтверджуючий сигнал має значення 1 («істина»).

На рис. 1.19, а наведена проста модель генератора ШІМ-імпульсів, складена на основі блока *Check Static Upper Bound*, а на рис. 1.19, б – осцилограми цієї моделі. Різниця двох сигналів з різними частотами – пилоподібного (1) і синусоїдного (2) – надходить на вхід *sig* блока верифікації, у налаштуваннях якого параметр

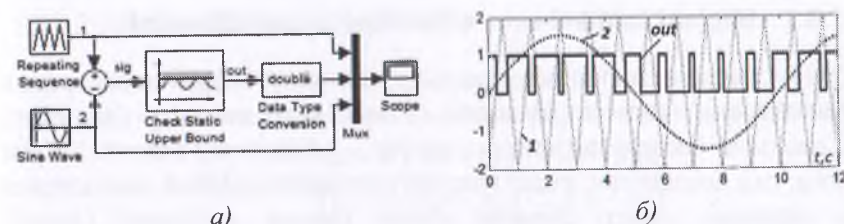


Рисунок 1.19 – Модель ШІМ-генератора з використанням блока *Check Static Upper Bound* (а) та графіки перехідного процесу (б)

Upper bound дорівнює 0 (значення верхньої границі), прапорець *Output assertion signal* установлений, а прапорець *Stop simulation...* знятий.

При таких налаштуваннях можна записати алгоритм роботи даного блоку:

$out = 1$, якщо $sig \leq 0$ (сигнал нижче заданої границі),
 $out = 0$, якщо $sig > 0$ (сигнал перевищує границю).

У результаті вихідний сигнал *out* має вигляд послідовності імпульсів, ширина яких змінюється за синусоїдним законом (докладніше принцип ШІМ розглянутий у п. 3.2.3, рис. 3.32). Даний приклад ілюструє використання блока верифікації при моделюванні систем керування.

Варто підкреслити, що у всіх блоків бібліотеки *Model Verification* сигнал на виході має тип boolean. Якщо цей сигнал використовується в моделі разом із сигналами інших типів, наприклад, при об'єднанні у вектор за допомогою блока *Mux*, то необхідно виконати перетворення сигналу (див. пояснення до блока *Hit Crossing*, п. 1.2.3.3).

1.3 Моделювання систем автоматичного керування

Для дослідження динамічних властивостей системи автоматичного керування (САК) необхідно мати її математичний опис (модель) у формі диференціальних рівнянь. Поряд з диференціальними рівняннями для опису систем часто використовуються передатні функції. Пакет *Simulink* надає широкі можливості моделювання як неперервних, так і дискретних систем, заданих у вигляді структурної схеми, диференціальних рівнянь, передатних функцій, простору станів. Пакет *Control System Toolbox* дає змогу досліджувати модель лінійної стаціонарної системи методами теорії автоматичного керування.

1.3.1 Моделі динамічних об'єктів у пакеті *Simulink*

Розглянемо, як реалізуються в *Simulink* моделі неперервних (аналогових) систем. Більшість з них знаходяться у бібліотеці *Continuous* (моделі дискретних систем – у бібліотеці *Discrete*). Крім того, ряд додаткових, рідко використовуваних моделей знаходиться в окремому пакеті *Simulink Extras* (розділ *Additional Linear*), розташованому на тому ж ієрархічному рівні браузеру бібліотек, що і *Simulink*.

1.3.1.1 Модель об'єкта у формі передатної функції *Transfer Fcn*

Передатна функція – це відношення зображень за Лапласом вихідної величини $y(s)$ до вхідного впливу $u(s)$ при нульових початкових умовах:

$$H(s) = \frac{y(s)}{u(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n},$$

де s – оператор Лапласа. При цьому степінь чисельника має бути не більше степеня знаменника: $m \leq n$.

У пакеті *Simulink* передатна функція лінійної системи реалізується блоком *Transfer Fcn* (рис. 1.20, а). У вікні налаштування блока в полях *Numerator* і *Denominator* задаються поліноми чисельника і знаменника передатної функції відповідно. У кожному полі в квадратних дужках вводиться рядок, який містить коефіцієнти полінома в порядку зменшення степеня оператора s . При цьому введена передатна функція відображається на піктограмі блока. Якщо його розміри не вміщують зображення повністю (рис. 1.20, б), то блок потрібно збільшити (розтягнути за допомогою покажчика миші). Як приклад на рис. 1.20, в, г показано блок *Transfer Fcn* із заданою передатною функцією та його вікно налаштування для даного випадку.

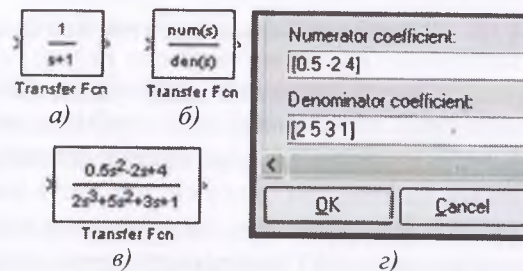


Рисунок 1.20 – Вікно налаштування і зовнішній вигляд блока *Transfer Fcn*

1.3.1.2 Модель об'єкта у формі «нулів-полісів» *Zero-Pole*

Блок *Zero-Pole* (рис. 1.21), являє собою модель лінійної системи, передатна функція якої задана у формі «нулів» і «полісів»:

$$H(s) = K \frac{(s - z_1) \cdot (s - z_2) \cdot \dots \cdot (s - z_m)}{(s - p_1) \cdot (s - p_2) \cdot \dots \cdot (s - p_n)},$$

де K – коефіцієнт підсилення; z_1, z_2, \dots, z_m (zeros) – корені полінома чисельника передатної функції, тобто значення, за яких $H(s) = 0$; p_1, p_2, \dots, p_n (poles) – корені полінома знаменника (характеристичного рівняння), тобто значення, за яких $H(s) = \infty$.

У вікні налаштування блока *Zero-Pole* корені чисельника і знаменника задаються в полях *Zeros* і *Poles* відповідно, у вигляді рядка значень у квадратних дужках. У полі *Gain* задається значення коефіцієнта підсилення. Наприклад, для передатної функції на рис. 1.21 треба ввести такі значення параметрів:

Zeros: [3 4]; *Poles*: [-2 8 -6]; *Gain*: 5

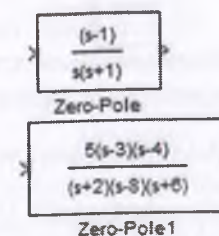


Рисунок 1.21 – Блок *Zero-Pole*

1.3.1.3 Модель об'єкта у формі простору станів *State-Space*

Блок *State-Space* (рис. 1.22) дає змогу задати математичний опис об'єкта у формі простору станів, тобто у вигляді системи диференціальних рівнянь:

$$\frac{dx}{dt} = A \cdot x + B \cdot u$$

$$y = C \cdot x + D \cdot u$$

де x – вектор станів системи; u – вектор вхідних сигналів; y – вектор вихідних сигналів; A, B, C, D – матриці коефіцієнтів. Для системи з кількістю станів n , що має m виходів і g входів, ці матриці мають такі розмірності:

$$A - [n \times n], B - [n \times g],$$

$$C - [m \times n], D - [m \times g].$$

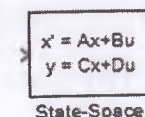


Рисунок 1.22 – Блок *State-Space*

Матриці A , B , C , D задаються в однойменних полях вікна налаштування блока *State-Space*. Значення коефіцієнтів кожної матриці вводяться в квадратних дужках, коефіцієнти одного й того ж рядка розділяються комою або пробілом, різні рядки розділяються між собою оператором «крапка з комою». Так, запис $[1, 2, 5; 0, 4, 12]$ дозволяє задати матрицю розмірністю $[2 \times 3]$:

$$\begin{bmatrix} 1 & 2 & 5 \\ 0 & 4 & 12 \end{bmatrix}$$

У полі *Initial conditions* вікна налаштування вводиться вектор початкових умов для всіх змінних стану системи. Кількість елементів вектора повинна відповідати кількості цих змінних.

1.3.1.4 Пропорційна ланка Gain

Блок *Gain* (бібліотека *Math Operations*), рис. 1.23, генерує вихідний сигнал y , який дорівнює добутку вхідного сигналу u на константу K :

$$y(t) = K \cdot u(t).$$

Значення коефіцієнта K задається в полі *Gain* вікна налаштування блока і відображається на його піктограмі. Вхідний сигнал u і коефіцієнт K можуть бути скалярними, векторними або матричними. Для нескаларних величин у полі *Multiplication* вікна налаштування можна вибрати спосіб перемножування: поелементний (*Element-wise*) або матричний (*Matrix*). Якщо значення константи не вміщається на піктограмі блока, то відображається символ $\langle -k- \rangle$. У цьому випадку розмір блока потрібно збільшити.

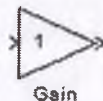


Рисунок 1.23 – Блок *Gain*

1.3.1.5 Диференціююча ланка Derivative

Передатна функція диференціюючої ланки

$$H(s) = K s,$$

де K – коефіцієнт підсилення.

Блок *Derivative* (рис. 1.24), генерує на виході похідну вхідного сигналу u за часом:

$$y(t) = \frac{du(t)}{dt}.$$

Коефіцієнт підсилення K прийнятий рівним 1. У вікні налаштування блока не передбачене задання цього коефіцієнта. Якщо потрібно інше значення K , можна послідовно з блоком *Derivative* включити блок *Gain*.

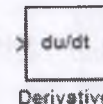


Рисунок 1.24 – Блок *Derivative*

У більш ранніх версіях програми блок *Derivative* не мав параметрів налаштування, у поточній версії він має єдиний параметр *Linearization Time Constant*, у полі якого задається значення сталої часу для лінеаризації даного блока шляхом додавання додаткового полюса в його передатну функцію. При додаванні полюса виникає ефект фільтрації сигналу перед його диференціюванням. За замовчуванням встановлене значення *Inf* (нескінченність), тобто лінеаризація не виконується.

1.3.1.6 Інтегруюча ланка Integrator

Вікно параметрів інтегратора наведене на рис. 1.25, а. Воно містить елементи, описані нижче.

External reset (зовнішнє скидання) – тип зовнішнього керуючого сигналу, що здійснює скидання вихідного сигналу блока в заданий початковий стан. Тип керуючого сигналу вибирається зі списку:

none – немає скидання;
rising – наростаючий сигнал;
falling – спадаючий сигнал;
either – будь-який сигнал.

Тип *rising* означає, що керуючий сигнал переходить з від'ємного або нульового значення в додатне,

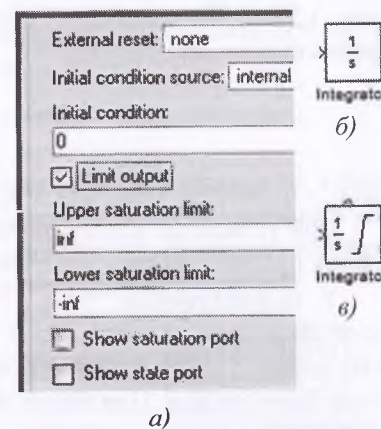


Рисунок 1.25 – Вікно налаштування та зовнішній вигляд блока *Integrator*

тип *falling* означає перехід з додатного значення в нуль або у від'ємне, тип *either* – перехід з нульового значення в ненульове або зміну знака.

Initial condition source – джерело початкового значення вихідного сигналу. У списку, що розкривається, можна вибрати внутрішнє (*internal*) або зовнішнє (*external*) джерело. У першому випадку в полі *Initial condition* вводять початкове значення сигналу, в другому випадку з'являється додатковий порт для підключення зовнішнього джерела.

Limit output – включення/відключення обмеження вихідного сигналу. При установці даного прапорця стають активними поля *Upper saturation limit* і *Lower saturation limit* для введення верхньої та нижньої границь обмеження (за замовчуванням задане значення *inf* – нескінченність).

Show saturation port – відображає вихідний порт, сигнал на якому приймає одне з трьох значень: 1 – вихідний сигнал блока досяг верхньої границі обмеження, -1 – досягнута нижня границя; 0 – сигнал знаходиться між заданими границями.

Show state port – керує відображенням порту стану системи, що використовується для розриву алгебраїчних циклів при моделюванні автоматичного скидання інтегратора в початковий стан.

Модель інтегруючої ланки представлена блоком *Integrator*, рис. 1.25, б. Його вихідний сигнал $y(t)$ дорівнює інтегралу від вхідного сигналу $u(t)$:

$$y(t) = \int_{t_0}^t u(t) dt + y_0,$$

де y_0 – початкове значення вихідного сигналу.

При установці всіх прапорців відображаються відповідні порти, і блок приймає вигляд, показаний на рис. 1.25, в.

У налаштуваннях блоку *Integrator* не передбачене введення коефіцієнта підсилення, тому, якщо потрібно реалізувати інтегруючу ланку з яким-небудь коефіцієнтом k , можна використовувати блок *Transfer Fcn*. При цьому його параметри треба задати такими: *Numerator* – $[k]$, *Denominator* – $[1; 0]$.

1.3.1.7 ПІД-ланка PID Controller

Блок *PID Controller* (бібліотека *Simulink Extras / Additional Linear*), рис. 1.26, дає змогу реалізувати різні варіанти ПІД-ланки, що часто використовуються в системах керування як регулятори.

Внутрішня модель блока являє собою паралельне з'єднання трьох складових – пропорційної, інтегральної та диференціальної. Вікно налаштування містить поля для введення значень коефіцієнтів кожної складової (*Proportional*, *Integral*, *Derivative*). Щоб виключити яку-небудь складову, необхідно установити її коефіцієнт у 0. Таким чином, за допомогою даного блока можуть бути реалізовані різні окремі випадки ПІД-ланки: пропорційно-інтегральна, пропорційно-диференціальна, інтегро-диференціальна.

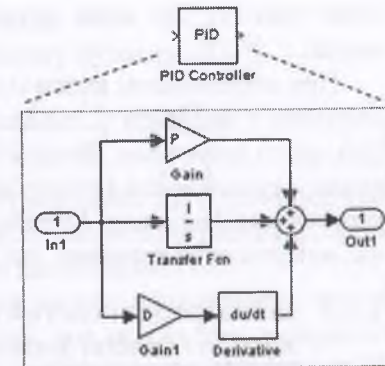


Рисунок 1.26 – Зовнішній вигляд та внутрішня модель блока *PID Controller*

1.3.1.8 Ланка із запізнюванням Transport Delay

Ланка із запізнюванням затримує вхідний сигнал $u(t)$ на заданий час τ . Вона описується рівнянням

$$y(t) = u(t - \tau).$$

Це означає, що в будь-який момент часу t сигнал y на виході дорівнює тому значенню вхідного сигналу u , яке він приймав у момент часу $(t - \tau)$. Іншими словами, сигнал на виході повністю повторює вхідний сигнал, але із запізнюванням на час τ .

Ланка із запізнюванням реалізується блоком *Transport Delay* (рис. 1.27). Затримка часу τ задається параметром *Time delay* вікна налаштування. Вихідний сигнал блока відповідає вхідному, зміщеному на час τ . При $t < \tau$ сигнал на виході блока приймає значення, задане в полі *Initial output*. У цей



Рисунок 1.27 – Блок *Transport Delay*

період часу блок зберігає значення вхідного сигналу в буфері, розмір якого визначається параметром *Initial buffer size*. Якщо кількість даних перевищує розмір буфера, то блок запитує додаткову пам'ять, і тоді *Simulink* видає повідомлення про те, який потрібен розмір буфера. При великому значенні параметра τ блоку потрібен великий обсяг пам'яті, що може суттєво сповільнити процес розрахунку моделі.

При моделюванні ланки із запізнюванням існує можливість час затримки τ подавати з зовнішньої моделі через спеціальний вхід. Для цього існує блок *Variable Time Delay* (рис. 1.27), який має два входи: верхній – для затримуваного сигналу, нижній – для сигналу керування τ . Параметр *Maximum delay* у вікні налаштування визначає максимальне значення, яке може приймати сигнал τ .

1.3.2 Дослідження систем автоматичного керування в пакеті *Control System Toolbox*

Розглянемо порядок складання моделей і дослідження їх засобами пакетів *Simulink* і *Control System Toolbox* на прикладі моделі двигуна постійного струму (ДПС) з незалежним збудженням. При керуванні напругою якоря $U_{\text{я}}$ вважаємо постійною напругу збудження $U_{\text{з}}$ і магнітний потік машини Φ . Рівняння, які описують електромагнітні й електромеханічні процеси в ДПС, відомі з теорії електропривода [1, 2]. Для заданих значень параметрів якірного кола ($R_{\text{я}}$, $L_{\text{я}}$) і моменту інерції J двигуна та елементів механізму вони мають вигляд:

$$U_{\text{я}} = R_{\text{я}} i_{\text{я}} + L_{\text{я}} \frac{di_{\text{я}}}{dt} + e,$$

$$J \frac{d\omega}{dt} = M - M_{\text{с}},$$

де $e = k\Phi\omega$ – ЕРС двигуна; $M = k\Phi i_{\text{я}}$ – його електромагнітний момент; ω – кутова частота обертання; $M_{\text{с}}$ – момент статичного навантаження; k – конструктивний коефіцієнт.

Запишемо ці рівняння в операторній формі:

$$U_{\text{я}} - e = i_{\text{я}}(R_{\text{я}} + L_{\text{я}}s), \quad (1.1)$$

$$M - M_{\text{с}} = J\omega s. \quad (1.2)$$

Знайшовши з (1.1) струм якоря $i_{\text{я}}$ і підставивши його в рівняння $M = k\Phi i_{\text{я}}$, одержимо передатну функцію $H_1(s)$:

$$H_1(s) = \frac{M}{U_{\text{я}} - e} = \frac{k\Phi}{R_{\text{я}}(T_{\text{я}}s + 1)},$$

де $T_{\text{я}} = L_{\text{я}}/R_{\text{я}}$ – стала часу якірного кола.

З рівняння (1.2) одержуємо передатну функцію $H_2(s)$:

$$H_2(s) = \frac{\omega}{M - M_{\text{с}}} = \frac{1}{Js}.$$

На рис. 1.28, а зображена структурна схема ДПС при постійному магнітному потоці, а на рис. 1.28, б – модель, складена з блоків, розглянутих у п. 1.3.1, відповідно до даної схеми.

Передатна функція $H_1(s)$ у даній моделі представлена трьома блоками – *Gain*, *Transfer Fcn* та *Gain1*, але можна було використувати тільки блок *Transfer Fcn* з відповідними коефіцієнтами. На вхід моделі подається постійний сигнал U від блока *Step1*, що моделює напругу якоря, від аналогічного блока *Step2* надходить сигнал впливу $M_{\text{с}}$. У даній моделі упродовж одного сеансу моделювання реалізуються два режими – пуск ДПС і накидання навантаження.

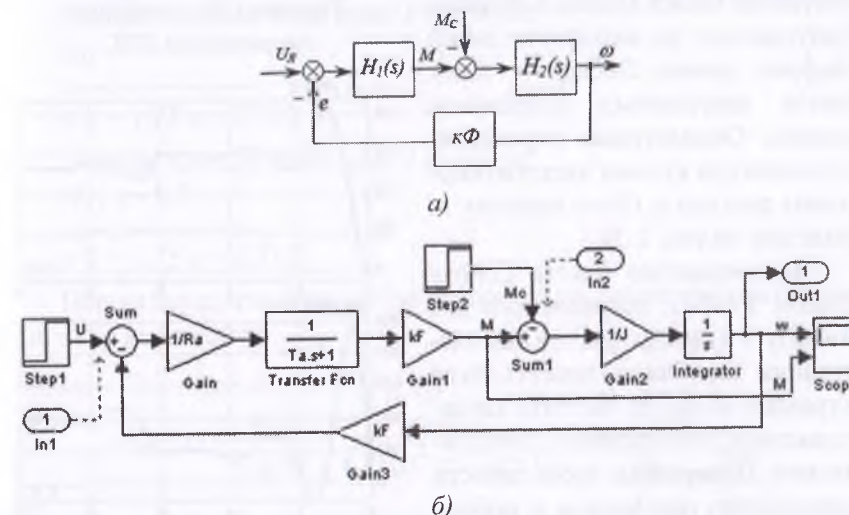


Рисунок 1.28 – Структурна схема (а) та модель (б) двигуна постійного струму

При цьому вхідний вплив подається відразу ж у момент часу $t = 0$ і не змінюється упродовж усього процесу розрахунку моделі. Збурення M_c спочатку дорівнює 200 Нм, а в момент часу $t = 6$ с стрибком зростає в 2 рази, тобто до 400 Нм.

Для реалізації вказаних режимів параметри джерел були встановлені у наступні значення:

блок *Step1*: *Step time* – 0; *Initial value* – 0; *Final value* – U ;

блок *Step2*: *Step time* – 6; *Initial value* – M_c ; *Final value* – $M_c \cdot 2$.

Якірне коло зі сталою часу T_a моделюється блоком *Transfer Fcn* з такими налаштуваннями параметрів: *Numerator* – [1]; *Denominator* – [Ta 1].

Параметри всіх елементів можна ввести в налаштування кожного блока безпосередньо в числовому вигляді, але зручніше використувати змінні, значення яких задані в m-файлі (див. п. 1.1.2). У даному випадку був сформований файл *param.m*, показаний на рис. 1.29. Він містить значення всіх змінних, імена яких введені в налаштування блоків моделі. Спочатку запускається на виконання даний m-файл (меню *Debug* → *Run*), потім запускається розрахунок моделі. Осцилограми перехідних процесів для кутової частоти обертання двигуна ω і його моменту M наведені на рис. 1.30.

За допомогою пакета *Control System Toolbox*, призначеного для аналізу і синтезу систем автоматичного керування, можуть бути отримані часові та частотні характеристики досліджуваної *Simulink*-моделі. Попередньо треба занести інформацію про модель в робочу область MATLAB у формі передатної функції. Для цього необхідно:

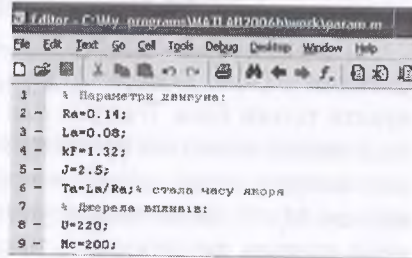


Рисунок 1.29 – m-файл з параметрами ДПС

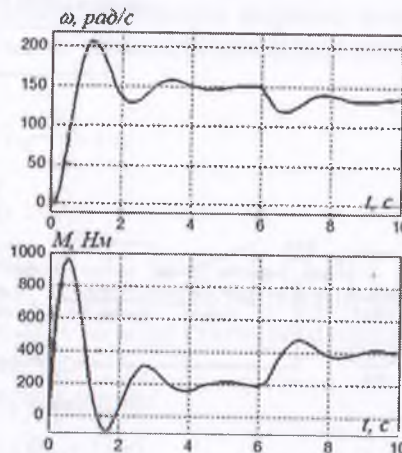


Рисунок 1.30 – Графіки перехідних процесів у моделі ДПС

1) до входу моделі замість джерела сигналу підключити блок *In* (бібліотека *Sources*), а до виходу – блок *Out* (бібліотека *Sinks*);

2) зберегти файл із моделлю під будь-яким іменем;

3) у командному вікні MATLAB ввести наступний запис:

```
[A,B,C,D] = linmod('file_name');
```

```
sys = ss(A,B,C,D);
```

```
h = tf(sys)
```

Тут *file_name* – довільне ім'я файлу моделі (без розширення .mdl), *linmod* – функція, що отримує інформацію із моделі *Simulink* у вигляді матриць A, B, C, D ; *ss* – команда, яка формує з цих матриць математичний опис системи з довільним іменем *sys* у формі простору станів; *tf* – команда перетворення системи з форми простору станів у форму передатної функції з довільним іменем *h*.

Кількість блоків *In* і *Out* визначає кількість входів і виходів системи відповідно. У досліджуваній моделі на рис. 1.28, б розглядаються два входи – за заданням та збуренням (блоки *In1* і *In2*, при цьому їх треба підключити замість блоків *Step1* і *Step2*), і один вихід – частота обертання двигуна ω (блок *Out1*). Таким чином, після введення зазначеної вище послідовності команд і натискання клавіші *Enter* у командному вікні з'являться дві передатні функції:

Transfer function from input 1 to output:

6.6

$$s^2 + 1,75s + 8,712$$

Transfer function from input 2 to output:

$$-0,4s - 0,7$$

$$s^2 + 1,75s + 8,712$$

Перша передатна функція (за заданням) є відношенням вихідного сигналу до вхідного:

$$H_{WU}(s) = \frac{\omega(s)}{U_{\text{я}}(s)} = \frac{1}{T_{\text{я}} T_{\text{м}} s^2 + T_{\text{м}} s + 1} = \frac{\frac{k\Phi}{L_{\text{я}} J}}{s^2 + \frac{R_{\text{я}}}{L_{\text{я}}} s + \frac{k\Phi^2}{L_{\text{я}} J}},$$

де $T_{\text{м}} = J R_{\text{я}} / (k\Phi)^2$ – електромеханічна стала часу.

Друга передатна функція (за збуренням) – відношення вихідного сигналу до збурюючого впливу:

$$H_{WM}(s) = \frac{\omega(s)}{M_C(s)} = -\frac{\frac{1}{(k\Phi)^2} R_J (T_J s + 1)}{T_J T_M s^2 + T_M s + 1} = -\frac{\frac{1}{J} s + \frac{R_J}{L_J J}}{s^2 + \frac{R_J}{L_J} s + \frac{k\Phi^2}{L_J J}}.$$

Для побудови часових і частотних характеристик системи використовуються наступні функції пакета *Control System Toolbox*:

Step – побудова перехідної функції;

Impulse – побудова імпульсної функції;

Bode – побудова логарифмічних амплітудно-частотної і фазочастотної характеристик (діаграма Бode);

Nyquist – побудова амплітудно-фазової характеристики (діаграма Найквіста).

Функцію потрібно ввести в командному вікні MATLAB, указавши після неї в дужках ім'я змінної, і натиснути *Enter*: наприклад, *step(h)*. Відкриється додаткове вікно з бажаною характеристикою системи. У даному випадку для моделі на рис. 1.28, б будуть побудовані дві характеристики, оскільки змінна *h* містить дві передатні функції – за задаванням і збуренням. Якщо потрібно досліджувати тільки одну з них, то замість імені змінної *h* вводиться запис *h(1)* або *h(2)* відповідно. Для відображення координатної сітки використовується команда *grid*. Для одержання детальної інформації про кожну з зазначених функцій треба ввести в командному вікні команду *help* та ім'я функції, наприклад *help step*.

1.4 Підсистеми

1.4.1 Прості підсистеми

При зростанні складності моделі виникає необхідність спрощення її шляхом об'єднання групи блоків у підсистеми. Використання підсистем має ряд переваг:

1) дає змогу об'єднати блоки в групи за функціональною ознакою і налаштовувати кожну групу окремо, після чого скласти основну модель з готових, функціонально завершених підмоделей;

2) підвищує наочність моделі шляхом зменшення кількості блоків, відображуваних у її вікні;

3) дає змогу користувачам створювати власну бібліотеку елементів, які мають вікна налаштування параметрів, піктограми, опис і т. п. (див. п. 1.4.3).

Будь-яка підсистема може містити в собі іншу підсистему, що дає змогу створювати ієрархічну структуру моделі. Для полегшення доступу до вкладених підмоделей використовується браузер моделей, що відкривається через меню *View → Model Browser Options → Model Browser* вікна моделі. При цьому в лівій частині вікна з'являється деревоподібна структура моделі (рис. 1.31), яка містить вкладені підсистеми. Вміст обраної підсистеми відображається в правій частині вікна. Основна модель знаходиться на верхньому рівні ієрархії.

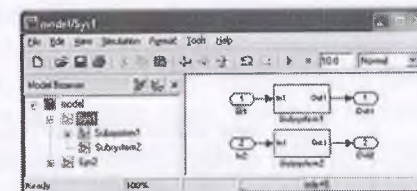


Рисунок 1.31 – Вікно моделі, що містить вкладені підсистеми

Для створення простих підсистем існує два способи.

Спосіб 1. Помістити у вікно моделі блок *Subsystem* (бібліотека *Ports and Subsystems*), потім відкрити його вікно і скласти в ньому потрібну підмодель. Для її зв'язку із зовнішньою моделлю використовувати вхідний і вихідний порти *In* і *Out*, які знаходяться в блоці *Subsystem*.

Спосіб 2. У готовій моделі виділити мишею групу блоків, які необхідно зібрати в підсистему, і вибрати в меню команду *Edit → Create Subsystem*. При цьому на місці виділених блоків з'явиться блок *Subsystem*, усередині якого буде поміщений обраний фрагмент моделі. На його входах і виходах автоматично з'являться блоки *In* і *Out*.

Кількість входів і виходів підсистеми визначається кількістю блоків *In* і *Out*. На піктограмі блока відображається номер порту. При додаванні цих блоків у підсистему їхні номери автоматично змінюються в порядку зростання. При необхідності також можуть бути змінені їхні імена, при цьому відповідно зміниться і маркування портів на піктограмі підсистеми.

За допомогою блока *Subsystem* можна створити у вікні моделі кнопку для запуску *m*-файлу, який реалізує певну функцію,

наприклад, побудову графіка. Для цього потрібно помістити в модель блок *Subsystem* і видалити його внутрішні порти *In* і *Out*. Далі, виділивши цей блок покажчиком миші, вибрати в меню *Edit* команду *Block Properties* і відкрити вкладку *Callbacks*. У вікні, що з'явилося, ліворуч вибрати функцію *OpenFcn*, а в поле праворуч ввести ім'я відповідного m-файлу (без розширення .m), у якому записані команди побудови графіка (див. п. 1.1.4). При цьому даний файл має знаходитися в поточній папці MATLAB (за замовчуванням це папка *Work*). Закрити вікно, натиснувши ОК. Тепер при подвійному натисненні лівої клавіші миші по блоку *Subsystem* буде виконуватися зазначений m-файл і здійснюватися побудова графіка.

1.4.2 Керовані підсистеми

Крім звичайних підсистем, у *Simulink* існують керовані підсистеми, які активізуються при наявності керуючого сигналу. Вони також знаходяться в бібліотеці *Ports and Subsystems*. Якщо керована підсистема активована, вона виконує обчислення і передає результати за допомогою вихідних портів *Out* у модель верхнього рівня. Якщо керована підсистема пасивна, то обчислення не виконуються.

1.4.2.1 E-підсистема Enabled Subsystem

Підсистема *Enabled Subsystem* (рис. 1.32, а) має керуючий вхід, на який надходить сигнал керування. Підсистема працює в тому випадку, якщо цей сигнал має додатне значення. Вона починає роботу на тому кроці моделювання, на якому сигнал перетинає нульове значення у додатному напрямку, і працює доти, поки він залишається

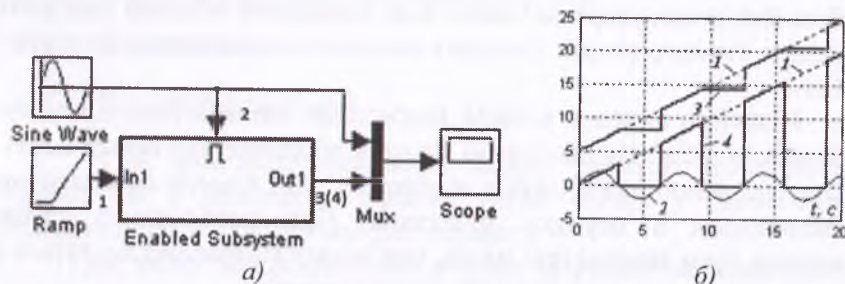


Рисунок 1.32 – Модель з використанням блока *Enabled Subsystem* (а) та графіки перехідного процесу (б)

додатним. У пасивному стані (при від'ємному сигналі керування) вихідний сигнал Е-підсистеми визначається налаштуваннями її вихідного порту *Out*. Блок *Out* у своєму вікні має параметр *Output when disabled* – значення на виході при забороненому стані. Він може бути встановлений в одне з двох значень:

held (утримання) – на вихід надходить сигнал, сформований при останньому спрацьовуванні підсистеми;

reset (скидання) – на вихід надходить сигнал, що відповідає початковому стану підсистеми; при цьому початковий стан задається в полі *Initial output*.

Таким чином, у пасивному стані Е-підсистема або затримує на виході попереднє значення, або скидає сигнал у початковий стан.

У своєму складі Е-підсистема, крім блоків *In* і *Out*, містить блок *Enable*, що нікуди не підключається, а лише задає ознаку її керованості. Якщо даний блок видалити, то керована підсистема перетвориться в звичайну. Прапорець *Show output port* у вікні налаштування блока *Enable* дає змогу відобразити вихідний порт, який може бути використаний для керування іншими блоками. У пасивному стані підсистеми сигнал на виході даного порту дорівнює 0, а в активному стані відповідає сигналу керування, що надходить на керуючий вхід підсистеми.

Як приклад розглянемо модель на рис. 1.32, а, осцилограми якої наведені на рис. 1.32, б. На робочий вхід *In1* Е-підсистеми подається лінійно-наростаючий сигнал 1 від блока *Ramp*, на керуючий вхід – синусоїда 2. В середині підсистеми вхід і вихід з'єднані безпосередньо, тобто в даному випадку це просто керована перемичка. Вихідний сигнал отриманий для двох випадків: якщо параметр *Output when disabled* блока *Out* встановлений у значення «*held*» (графік 3) або в значення «*reset*» (графік 4). У полі *Initial output* блока *Out* встановлено значення 0. Видно, що підсистема працює лише в додатні напівперіоди керуючої напруги, а у від'ємні напівперіоди або зберігає значення з попереднього запуску (3), або скидає сигнал у нуль (4). У налаштуваннях блока *Ramp* для кожного випадку задавалися різні початкові значення сигналу (5 і 0 відповідно), тому графік 1 представлений у двох варіантах.

Інший приклад застосування блока *Enabled Subsystem* – підсистема для визначення максимального значення вхідного сигналу. Незважаючи на багатий вибір функцій, у *Simulink* немає блока, що

реалізує пошук екстремуму. Модель на рис. 1.33, а визначає максимальне значення вхідного сигналу, а також момент часу, що відповідає цьому значенню. Модель реалізована у вигляді підсистеми *Subsystem*, до складу якої входить керована підсистема *Enabled Subsystem* (рис. 1.34, а). Всередині неї два вхідних порти *In1* і *In2* з'єднані перемичками з відповідними вихідними портами *Out1* і *Out2* (рис. 1.34, б).

Параметри блоків *Out* встановлені в значення «held». Вхідний сигнал, що надходить із зовнішньої моделі, проходить через блок *Memory* (бібліотека *Discrete*), який «запам'ятовує» цей сигнал і затримує його на один крок часу. Тобто на виході блока *Memory* одержуємо попереднє значення x_{i-1} вхідної величини x_i . Далі значення x_{i-1} надходить на Е-підсистему і на її виході порівнюється з поточним значенням x_i цього ж сигналу. Їхня різниця $\Delta x = x_i - x_{i-1}$ як сигнал керування подається на керуючий вхід Е-підсистеми. Якщо $\Delta x > 0$ (вхідний сигнал зростає), то підсистема знаходиться в активному стані і працює як перемичка, передаючи на вихід значення x_{i-1} . Як тільки досягнуто максимум x_{\max} і починається спадання вхідного

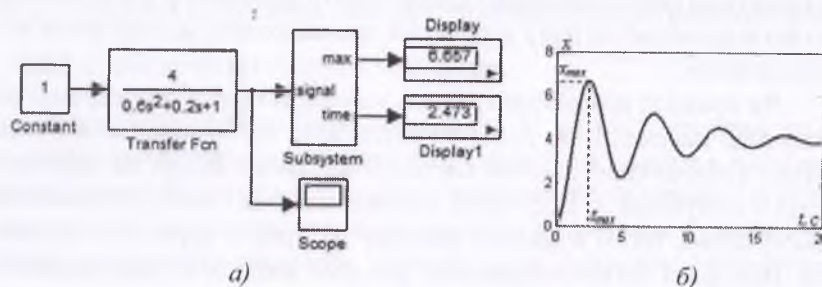


Рисунок 1.33 – Модель пошуку екстремуму (а) та осцилограма вихідного сигналу (б)

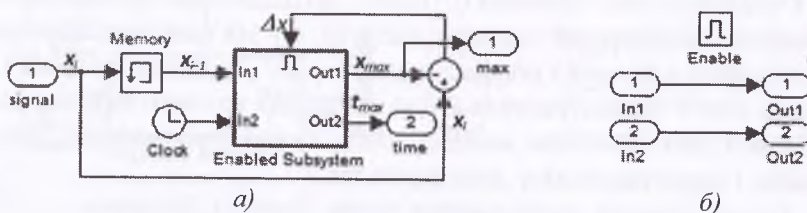


Рисунок 1.34 – Внутрішні моделі блоків *Subsystem* (а) та *Enabled Subsystem* (б)

сигналу ($\Delta x < 0$), Е-підсистема припиняє роботу і зберігає на своєму виході останнє значення, що надійшло в неї, тобто $x_{i-1} = x_{\max}$. Надалі, якщо сигнал знову починає зростати (коливальний процес), тоді кожне нове значення порівнюється зі збереженим максимальним x_{\max} на виході підсистеми, і якщо новий екстремум $x_{\max 1}$ буде більше попереднього x_{\max} , то підсистема активується і зберігає вже нове значення $x_{\max 1}$.

Аналогічно визначається і час досягнення максимуму t_{\max} . Для цього в Е-підсистемі існує друга перемичка. Генератор системного часу *Clock* (бібліотека *Sources*) передає в підсистему значення часу тільки в ті моменти, коли вона активна. В інші моменти підсистема зберігає останнє отримане нею значення t_{\max} .

У результаті знайдені значення x_{\max} і t_{\max} виводяться з підсистеми через виходи, позначені відповідно *max* і *time* (блоки *Out* зі зміненими іменами) і відображаються за допомогою блоків *Display*. На графіку на рис. 1.33, б видно, що ці значення відповідають максимуму сигналу і часу його досягнення.

1.4.2.2 Т-підсистема *Triggered Subsystem*

Наступний вид керованих підсистем – блок *Triggered Subsystem* (рис. 1.35, а). Він також має керуючий вхід. Т-підсистема працює лише на тому кроці моделювання, на якому відбулася зміна полярності керуючого сигналу. При цьому вона не повертається у початковий стан, і її поточний стан зберігається до чергового запуску. У вікні блока *Triggered Subsystem*, крім вхідного і вихідного портів, знаходиться блок *Trigger*, що визначає умови спрацьовування підсистеми. У його вікні параметр *Trigger type* може бути встановлений в одне з наступних значень:

rising – зміна полярності керуючого сигналу в додатному напрямку;

falling – зміна полярності у від'ємному напрямку;

either – зміна полярності в обох напрямках;

function-call – виклик заданої S-функції.

У залежності від обраного значення змінюється зовнішній вигляд піктограми блока *Trigger*. Другим параметром налаштування даного блока є прапорець *Show output port*, що використовується аналогічно з однойменним параметром блока *Enable*.

До складу Т-підсистеми можуть входити будь-які блоки пакета *Simulink*, за умови, що в них параметр еталонного часу *Sample time* дорівнює або нескінченності (*inf*), або -1 (*inherited*, або «наслідуваний» від моделі верхнього рівня). Якщо в налаштуваннях блока є прапорець *Inherit sample time*, то він має бути встановлений. Якщо блок не має параметра *Sample time* у явному вигляді, але фактично цей параметр відрізняється від -1 (як, наприклад, у блока *Integrator* та інших з бібліотеки *Continuous*), то використання такого блока в Т-підсистемі приведе до повідомлення про помилку.

Розглянемо як приклад модель на рис. 1.35, а. Всередині підсистеми *Triggered Subsystem* вхід і вихід з'єднані перемичкою. Параметр *Trigger type* блока *Trigger* встановлений у значення *either*. Осцилограми наведені на рис. 1.35, б для наступних сигналів: 1 – вхідний, 2 – керуючий, 3 – сигнал на виході Т-підсистеми. Видно, що її спрацювання відбувається в момент переходу керуючого сигналу через нуль як в одному, так і в іншому напрямку. У ці моменти підсистема пропускає на вихід вхідний лінійно-наростаючий сигнал. У періоди між спрацюваннями вона зберігає на виході останнє отримане значення.

Ще один приклад з використанням тригерної підсистеми – модель приладу для виміру частоти періодичного сигналу (рис. 1.36). Вона може бути корисна при моделюванні електричних кіл та електромеханічних систем за допомогою пакета *Sim Power Systems* (розділи 2–4). Синусоїдний сигнал, частоту якого потрібно виміряти, надходить на керуючі входи двох блоків *Triggered Subsystem*. Всередині кожної підсистеми вхід і вихід з'єднані перемичкою. На

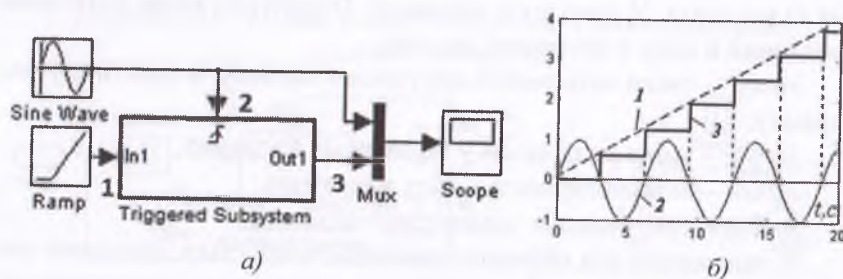


Рисунок 1.35 – Модель з використанням блока *Triggered Subsystem* (а) і графіки перехідного процесу (б)

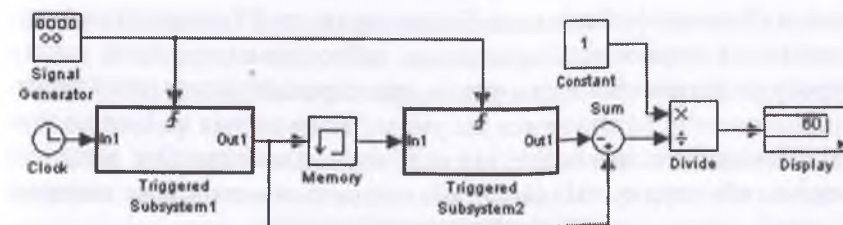


Рисунок 1.36 – Модель частотомира

робочий вхід першої підсистеми надходить поточний час моделювання від блока *Clock*, далі на її виході сигнал «запам'ятовується» блоком *Memory* і надходить на другу підсистему.

У момент переходу керуючого сигналу через 0 у додатному напрямку (параметр *Trigger type* встановлений у значення *rising*) обидві підсистеми спрацьовують і пропускають на вихід вхідний сигнал: для першої підсистеми – це поточний час моделювання t_i від блока *Clock*, для другої підсистеми – це значення часу t_{i-1} з попереднього кроку моделювання, що «запам'ятовується» блоком *Memory*. Сигнали на виходах залишаються незмінними до наступного спрацювання підсистем, тому час t_i – це момент останнього переходу синусоїди через 0, а час t_{i-1} – це момент передостаннього переходу. Далі з першого значення віднімається друге, і в результаті виходить різниця $\Delta t = t_i - t_{i-1}$, яка фактично являє собою період синусоїдного сигналу. Частота цього сигналу f визначається за допомогою блока *Divide*, який відповідно до виразу $f = 1/\Delta t$ виконує ділення сигналу першого входу (1) на сигнал другого входу (Δt). Виміряне значення частоти (60 Гц), яке відображується блоком *Display*, відповідає заданому значенню в налаштуваннях блока *Signal Generator*.

1.4.2.3 ET-підсистема Enabled and Triggered Subsystem

Третій вид керованих підсистем – блок *Enabled and Triggered Subsystem*. Він поєднує в собі властивості двох розглянутих вище підсистем. У своєму складі, крім перемички, він містить обидва блоки – *Enable* і *Trigger*, і, відповідно, має два керуючих входи. Алгоритм роботи ЕТ-підсистеми полягає в наступному. Якщо на деякому кроці моделювання відбувається зміна полярності «тригерного» керуючого сигналу в заданому напрямку (*rising*, *falling* або *either*), то програма перевіряє значення сигналу на іншому керуючому

вході (Е-сигнал). Якщо воно більше нуля, то ЕТ-підсистема запускається і працює як Т-підсистема, тобто тільки протягом даного кроку моделювання. Таким чином, для спрацювання даної підсистеми повинні виконуватися дві умови. Коли сигнал на Е-вході стає від'ємним, то в залежності від налаштування блока *Out*, вихідний сигнал або утримується (*held*), або скидається в початкове значення (*reset*).

Як приклад розглянемо модель, що реалізує амплітудно-імпульсну модуляцію сигналу (рис. 1.37, а). Осцилограми зображені на рис. 1.37, б. На робочий вхід блока *Enabled and Triggered Subsystem* подається модулюючий сигнал 1 низької частоти $\Omega_1 = 1$ рад/с. На обидва керуючих входи надходить сигнал 2 високої частоти $\Omega_2 = 20$ рад/с (синхронізуючий). Коли сигнал 2 змінює полярність у додатному напрямку, то виконуються обидві умови, і підсистема активується, пропускаючи на вихід сигнал 1. Отримане значення зберігається упродовж додатного напівперіоду сигнала 2, а потім скидається в 0 (у налаштуваннях блока *Out* задані параметри: «*reset*», *Initial output* = 0).

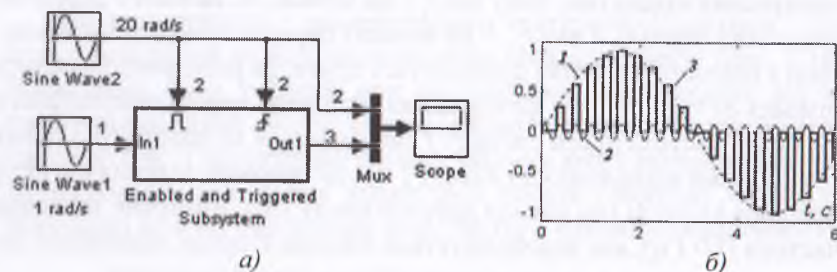


Рисунок 1.37 – Модель амплітудно-імпульсного модулятора (а) і графіки перехідного процесу (б)

У результаті вихідний сигнал 3 являє собою послідовність імпульсів, амплітуда яких змінюється за синусоїдним законом.

1.4.2.4 Керована за умовою підсистема *Action Subsystem*

Ця підсистема призначена для роботи під керуванням блоків *If* або *Switch Case*. У першому випадку вона називається *If Action Subsystem*, у другому – *Switch Case Action Subsystem*. Ознакою її керуваності є блок *Action Port*, що забезпечує наявність керуючого

входу. Параметри підсистеми визначаються налаштуваннями *held* і *reset* її вихідного порту *Out*, які були розглянуті при описі блока *Enabled Subsystem*. Блок активується при наявності сигналу дозволу на керуючому вході *Action*.

Блок умовного оператора *If* (рис. 1.38, а) призначений для керування підсистемою *If Action Subsystem*. Він є аналогом оператора *if-else* мови програмування С. Вікно налаштування блока *If* (рис. 1.38, б) містить наступні поля:

Number of inputs – кількість входів.

If expression – умовний вираз. Він може містити наступні оператори: $<$, $<=$, $=$ (дорівнює), \sim (не дорівнює), $>$, $>=$, $\&$ (логічне «і»), $|$ (логічне «або»), \sim (логічне «не»). Вхідний сигнал позначається *u1*. Якщо входів у блока декілька (параметр *Number of inputs* більше 1), то наступні сигнали позначаються *u2*, *u3* і т. д. Якщо вхідні сигнали векторні, то для позначення елементів вектора використовуються вирази *u1(1)*, *u1(2)*, *u2(1)*, *u2(2)* і т. д. На вихідному *If*-порту блока формується сигнал дозволу у тому випадку, якщо записаний умовний вираз є істинним.

Elseif expressions – один або декілька альтернативних умовних виразів, які розділені комами, і обчислюються в тому випадку, якщо умовний вираз *If expression* є хибним. Кожному умовному виразу, записаному в полі *Elseif expressions*, відповідає вихідний *Elseif-norm*, на якому формується керуючий сигнал, якщо відповідний умовний вираз є істинним. При цьому алгоритм обчислення альтернативних умовних виразів такий, що якщо один з них виявиться істинним,

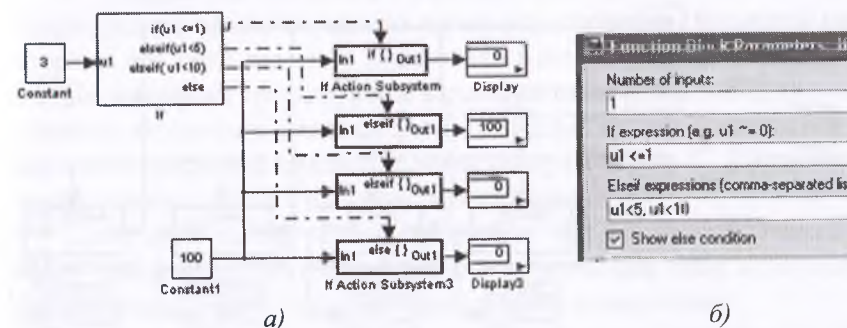


Рисунок 1.38 – Модель з використанням блока *If* (а) та його вікно налаштування (б)

то наступні в списку вирази не перевіряються. Дані вирази можуть містити в собі ті ж знаки, що і вирази в полі *If expression*.

Show else condition (прапорець) – показати *Else-norm*. На *Else-portu* формується керуючий сигнал, якщо умовний вираз і всі альтернативні умовні вирази хибні.

Піктограма блока відображає усі вирази, які записані в його параметрах. Додавання кожного нового альтернативного умовного виразу призводить до появи нового *Elseif*-порту.

На рис. 1.38, а показаний приклад використання блока *If* разом з підсистемами *If Action Subsystem*, у кожній з яких вхід і вихід з'єднані перемичкою. У цьому прикладі перша підсистема пропускає через себе вхідний сигнал, рівний 100, якщо сигнал *u1* на вході блока *If* менший або дорівнює 1 ($u1 \leq 1$), друга підсистема – якщо $1 < u1 < 5$, третя – якщо $5 \leq u1 < 10$, і четверта – у всіх інших випадках ($u1 \geq 10$). Оскільки в даній моделі $u1 = 3$, то працює друга підсистема.

Блок перемикача *Switch Case* (рис. 1.39) забезпечує формування керуючих сигналів для підсистеми *Case Action Subsystem*. Блок є аналогом оператора *Switch* мови програмування C. Вікно його налаштування містить поле *Case conditions* – список значень вхідних сигналів *u1* (ціле число). Кожному значенню відповідає окремий вихідний *Case*-порт. Якщо значення вхідного сигналу, що надходить на вхід *u1* блока *Switch Case*, збігається з яким-небудь значенням зі списку, то на відповідному виході блока формується керуючий сигнал. Якщо вхідний сигнал не є цілим, то його дробова частина

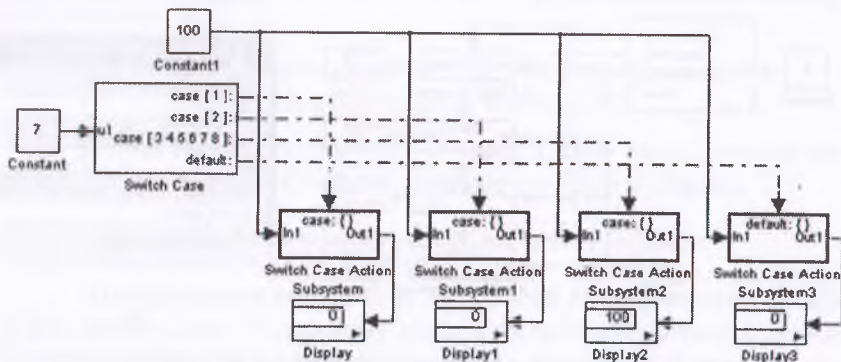


Рисунок 1.39 – Модель з використанням блока *Switch Case*

відкидається. У полі *Case conditions* можна використовувати квадратні дужки, якщо необхідно формувати керуючий сигнал на якому-небудь порту для кількох значень вхідного сигналу. Наприклад, вираз $\{1,[3,5]\}$ задає два вихідних *Case*-порти. На першому з них керуючий сигнал формується, якщо вхідний сигнал *u1* блока дорівнює 1, а на другому – якщо $u1 = 3$ або $u1 = 5$. У полі *Case conditions* можна використовувати також діапазони значень. Наприклад, вираз $\{3:8\}$ означає, що керуючий сигнал згенерується, якщо значення сигналу *u1* знаходиться в діапазоні від 3 до 8 включно.

Show default case (прапорець) – показати *default-norm*. На виході цього порту формується керуючий сигнал, якщо вхідний сигнал блока не збігається з жодним значенням, перерахованим у списку *Case conditions*.

На рис. 1.39 показаний приклад використання блока *Switch Case* разом з підсистемами *Switch Case Action Subsystem*, у кожній з яких вхід і вихід з'єднані перемичкою. У прикладі перша підсистема пропускає через себе сигнал, рівний 100, якщо вхідний сигнал блока *Switch Case* $u1 = 1$, друга підсистема – якщо $u1 = 2$, третя – якщо $3 \leq u1 \leq 8$, і четверта – у всіх інших випадках. У даній моделі $u1 = 7$, тому працює третя підсистема. При цьому в полі *Case conditions* блока *Switch Case* введений рядок $\{1,2,[3:8]\}$, а прапорець *Show default case* встановлений.

1.4.2.5 Підсистема циклу *For Iterator Subsystem*

Підсистема *For Iterator Subsystem* (рис. 1.40, а), виконується неодноразово протягом одного кроку моделювання. Кількість повторень має бути відомою заздалегідь і може задаватися або зовнішнім джерелом сигналу, або в налаштуваннях блока. Основні властивості підсистеми визначає її внутрішній блок *For Iterator*. Він є аналогом оператора циклу *For* мови програмування C. Його вікно налаштування містить наступні основні параметри:

States when starting – стан підсистеми при наступному запуску: *held* – використовувати попередній стан (останній стан, коли вона була активна), *reset* – використовувати початковий стан;

Iteration limit source (або *Source of number of iterations*) – визначає джерело кількості ітерацій: *internal* – внутрішнє, *external* – зовнішнє. Якщо обрано значення *external*, то в блоці *For Iterator*

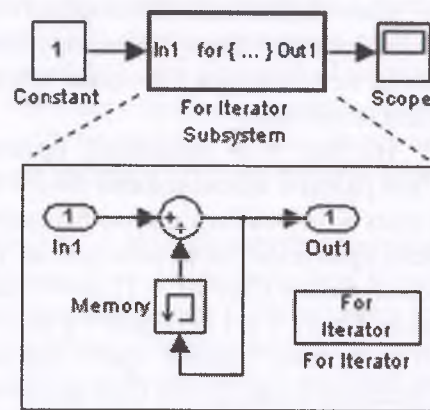
з'являється вхідний порт, на який надходить сигнал від зовнішнього джерела.

Причому саме джерело має знаходитися в моделі верхнього рівня і подавати сигнал на *For Iterator* через блок вхідного порту *In*;

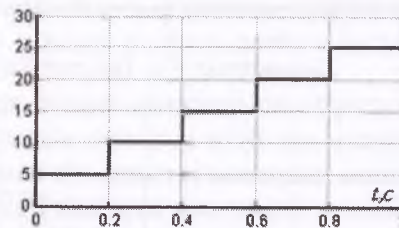
Iteration limit (або *Number of iterations*) – кількість ітерацій. Параметр доступний, якщо обрано внутрішнє джерело;

Show iteration number port – відобразити на піктограмі блока вихідний порт, з якого знімається сигнал номера ітерації.

На рис. 1.40, а показаний приклад використання блока *For Iterator Subsystem*. У прикладі виконується накопичення суми значень з кроком, рівним 1. Кількість ітерацій встановлена всередині блока *For Iterator* і дорівнює 5. Після закінчення заданої кількості ітерацій сигнал з підсистеми передається в зовнішню модель. У даному випадку в налаштуваннях моделі встановлено режим моделювання з фіксованим кроком, який дорівнює 0.2 (див. п. 1.1.3), а тривалість моделювання становить 1 с. Таким чином, модель за час 1 с виконує 5 кроків і, отже, підсистема *For Iterator Subsystem* здійснює 5 повних циклів розрахунку своєї внутрішньої моделі (приріст сигналу). Після закінчення кожного кроку часу сигнал надходить у зовнішню модель, і в результаті формується ступінчастий графік, показаний на рис. 1.40, б.



а)



б)

Рисунок 1.40 – Модель з використанням блока *For Iterator Subsystem* (а) і графік на виході моделі (б)

1.4.2.6 Підсистема циклу *While Iterator Subsystem*

Підсистема *While Iterator Subsystem*, як і попередня, також виконується неодноразово протягом одного кроку моделювання. Зовнішній вигляд блока і його внутрішній вміст зображені на рис. 1.41. Кількість повторень заздалегідь не відома.

Робота підсистеми відбувається увесь час, поки значення логічного сигналу на керуючому вході *cond* її внутрішнього блока *While Iterator* дорівнює *True* (Істина). Джерело керуючого сигналу *cond* має знаходитися всередині підсистеми. Цикл припиняється, коли цей сигнал приймає значення *False* (Хибно). Блок *While Iterator* є аналогом оператора циклу *while* (*do-while*) мови програмування C. Його основні параметри:

Maximum number of iterations – максимальна кількість ітерацій. Якщо значення параметра дорівнює -1, то кількість ітерацій не обмежується. Розрахунок підсистеми повторюється доти, поки значення *cond*-сигналу істинне і кількість ітерацій не перевищує задане максимальне значення.

While loop type – тип циклу. Вибирається зі списку:

- *while* – у цьому режимі в блоці *While Iterator* присутні два входи (див. рис. 1.41) – *cond* та *IC* (початковий стан). На початку поточного кроку моделювання програма перевіряє значення сигналу на *IC*-вході, і якщо воно істинне, то виконується розрахунок підсистеми, а потім перевіряється сигнал на керуючому вході *cond*. Якщо *IC*-сигнал дорівнює нулю, то на даному кроці розрахунок не виконується;
- *do-while* – у цьому режимі блок має тільки керуючий вхід *cond*. Спочатку виконується розрахунок підсистеми, потім перевіряється умова на вході *cond*.

States when starting – стан підсистеми при наступному запуску: *held* – використовувати попередній стан, *reset* – використовувати початковий стан.

Show iteration number port – відобразити вихідний порт, з якого знімається сигнал номера ітерації.

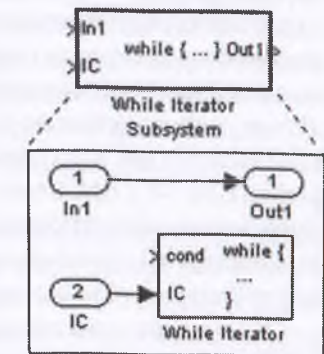


Рисунок 1.41 – Блок *While Iterator Subsystem*

1.4.3 Масковані підсистеми

Незважаючи на різноманітність стандартних бібліотечних елементів пакета *Simulink*, розроблювач може зіткнутися з необхідністю створення власних блоків, які мають такі ж властивості, як і стандартні елементи – вікно налаштування параметрів, піктограму, текстову документацію. Для цього застосовуються масковані підсистеми, зміст яких приховано спеціальним графічним інтерфейсом – маскою. При цьому, на відміну від звичайної, маскована підсистема сприймається користувачем не як набір блоків, об'єднаних виконуваною функцією, а як цільний блок із власною піктограмою і вікном налаштування параметрів. Більшість бібліотечних блоків пакета *Sim Power Systems* (розділи 2–4) є маскованими підсистемами. При необхідності їхній внутрішній зміст може бути відкрито за допомогою меню *Edit* → *Look under mask* вікна моделі (попередньо блок має бути виділений курсором миші). Таким чином, застосування маски спрощує налаштування елементів підсистеми, а також захищає її внутрішню модель від ненавмисної модифікації.

Як приклад розглянемо модель простої підсистеми, що реалізує обчислення вихідного сигналу відповідно до рівняння $y = ax + b$, де x – сигнал на вході, a і b – коефіцієнти. Підсистема та її внутрішня модель наведені на рис. 1.42 (для більшої наочності ім'я блока *Subsystem* змінено на « $ax + b$ »). Коефіцієнти блоків *Gain* і *Constant* задані у вигляді змінних a і b , значення яких вводяться у вікні налаштування створеної маски (рис. 1.43).

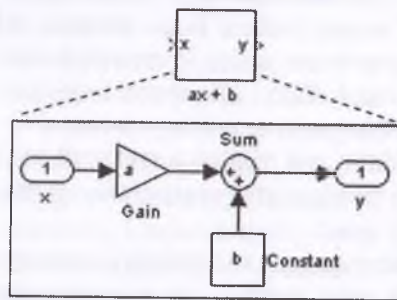


Рисунок 1.42 – Блок « $ax + b$ » та його внутрішня модель

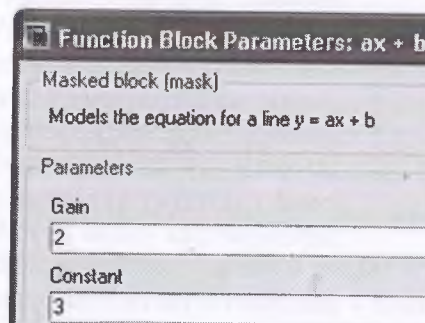


Рисунок 1.43 – Вікно маски блока « $ax + b$ »

Для маскування підсистеми використовується спеціальний редактор *Mask Editor*, що запускається за допомогою меню *Edit* → *Mask subsystem* (попередньо треба виділити підсистему покажчиком миші). Вікно редактора містить чотири розділи. Розділ *Parameters* (рис. 1.44) застосовується для підготовки вікна налаштування параметрів підсистеми. При натисканні кнопки (*Add*) з'являється додатковий рядок для введення наступних даних: *Prompt* – найменування параметрів у вікні налаштування, *Variable* – імена змінних, які мають значення відповідних параметрів, *Type* – тип параметрів. Параметри, що задаються, можуть бути наступних типів: *Edit* – поле для введення значення, *Checkbox* – прапорець, *Popup* – випадаючий список. Встановлений прапорець *Evaluate* означає, що даний параметр є обчислюваним (в іншому випадку він є символьним), а установка прапорця *Tunable* дає змогу змінювати значення параметра в процесі розрахунку моделі.

Для створення текстового опису використовується розділ *Documentation*. Він дає змогу задати три види опису: *Mask type* – найменування блока, після якого автоматично додається слово «(mask)» (рис. 1.43), щоб відрізнити масковану підсистему від стандартного бібліотечного блока; *Mask description* – опис блока, розміщений у верхній частині вікна установки параметрів (на рис. 1.43 у даному розділі введена фраза «*Models the equation for a line y = ax + b*»); *Mask help* – опис блока, розміщений у довідковій системі, що викликається натисканням кнопки *Help*.

Розділ *Initialization* редактора маски призначений для введення команд, що виконуються при ініціалізації моделі. Ініціалізація (установка початкових значень) здійснюється при завантаженні моделі, запуску моделювання або повороті блока. Якщо в полі ініціалізації задати певні значення змінних, наприклад, $a = 1$, $b = 0$, то

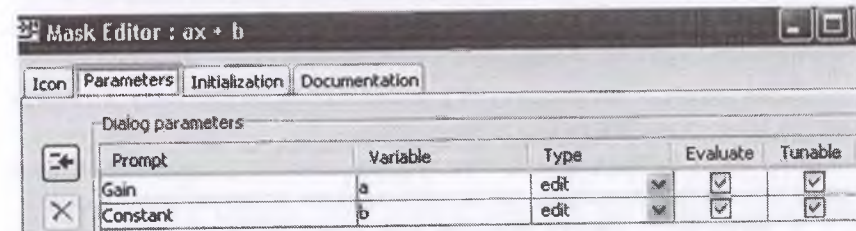


Рисунок 1.44 – Вікно редактора маски *Mask Editor*

вони будуть використовуватися кожен раз при запуску моделювання, незалежно від того, які значення введені у вікні налаштування блока. Якщо ніяких команд при ініціалізації виконувати не потрібно, то дане поле рекомендується залишити порожнім.

Розділ *Icon* дає змогу створити піктограму блока, яка містить текст, формули, зображення, графіку. Поле *Drawing commands* призначене для введення відповідних команд мовою програмування MATLAB. Для спрощення процесу підготовки графічних зображень застосовується редактор піктограм, що запускається командою *iconedit* (вона вводиться після курсора у вікні MATLAB і супроводжується натисканням клавіші *Enter*). Програма запитує ім'я файлу (без розширення) та ім'я блока, і потім, після введення імен, відкривається вікно редактора піктограм. Воно являє собою графічне вікно з координатною сіткою, на якому за допомогою миші можна зобразити будь-яку фігуру. Після виходу з редактора на піктограмі блока з'явиться створене зображення, а в полі *Drawing commands* розділу *Icon* з'явиться команда, що відповідає даному зображенню. Крім того, піктограму можна помістити зображення з будь-якого графічного файлу форматів PCX, JPG, TIF, BMP та ін. Для цього в полі *Drawing commands* потрібно ввести команду

```
image(imread('file_name'));
```

де *file_name* – ім'я графічного файлу (з розширенням). Даний файл має знаходитись у поточній папці MATLAB.

Якщо надалі виникає необхідність відредагувати створену маску, то повернутися до редактора маски можна за допомогою команди меню *Edit* → *Edit Mask*.

Контрольні питання до розділу 1

1. Подайте алгоритм створення моделей у пакеті *Simulink*.
2. У яких випадках доцільно задавати параметри моделі за допомогою *m*-файлу? Як працювати з *m*-файлом?
3. Як змінити час розрахунку моделі (крок розрахунку, точність, метод моделювання та ін.) у *Simulink*?
4. У яких випадках потрібно вручну змінити розмір максимального кроку моделювання, за замовчуванням встановлений в *auto*?
5. Який блок пакета *Simulink* дає змогу побудувати графік залежності $y = f(x)$, де x та y є функціями часу?

6. Як здійснити запис результатів моделювання в робочу область MATLAB? З якою метою це виконується?

7. З якою метою застосовується блок *Switch*? Пояснити алгоритм його роботи.

8. Дати характеристику блока *Relay* і пояснити алгоритм його роботи.

9. Для чого застосовуються блоки верифікації сигналу?

10. Як у командному вікні MATLAB одержати передатну функцію системи, модель якої складена в *Simulink*?

11. Як реалізувати модель ланки із запізнюванням? Які існують варіанти задання часу запізнювання?

12. Для яких цілей застосовуються керовані підсистеми?

13. Чим відрізняються алгоритми роботи підсистем *Enabled* і *Triggered*?

14. Чим визначається кількість ітерацій для блока *For Iterator Subsystem*? Для блока *While Iterator Subsystem*?

15. З якою метою в *Simulink* застосовується редактор маски *Mask Editor*? Які його можливості?

Завдання для самостійного виконання

1. За допомогою блоків *Repeating Sequence* і *Saturation* скласти модель джерела сигналу трапецеїдальної форми. Результати моделювання записати в робочу область.

2. На виході блока *Sine wave* одержати три синусоїдних сигнали, однакових за амплітудою і зміщених за фазою один відносно іншого на 120° .

3. Скласти модель генератора ШІМ-імпульсів, що за принципом дії аналогічна наведеним на рис. 1.19, але замість блока *Check Static Upper Bound* використовувати блок *Switch*.

4. Виконати завдання п. 3, але з застосуванням блоків *If* та *If Action Subsystem*.

5. Скласти моделі елементів, заданих у вигляді передатних функцій:

$$H_1(s) = \frac{20}{s^2}; \quad H_2(s) = \frac{s+1}{s^3+0,5}; \quad H_3(s) = \frac{s-2}{s(s+0,01)(s+0,4)}.$$

6. Скласти модель інтегруючої ланки з від'ємним зворотним зв'язком. Коефіцієнт підсилення в колі зворотного зв'язку дорівнює 2. Початкове значення сигналу на виході інтегратора дорівнює 0,5.

Розділ 2

МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ КІЛ ТА ЇХ ЕЛЕМЕНТІВ

2.1 Особливості застосування пакетів *Sim Power Systems* і *Simulink* для моделювання електротехнічних пристроїв

Для моделювання систем конкретного призначення – електротехнічних та енергетичних об'єктів – існує спеціальний пакет *Sim Power Systems (SPS)*, що є частиною комплексу програм середовища MATLAB. Серед застосування пакета *SPS* дуже широка – від аналізу окремих електричних кіл до моделювання ліній електропередачі, систем електроприводів та складних процесів у потужних енергетичних системах. Пакет *SPS* працює на базі *Simulink* і органічно взаємодіє з ним, використовуючи те ж програмне середовище, графічний інтерфейс і браузер бібліотек. Тому при моделюванні електроенергетичних пристроїв розроблювач має змогу використовувати переваги обох пакетів. Розглянемо основні особливості взаємодії *Simulink* і *Sim Power Systems*.

Simulink призначений для моделювання динамічних систем, заданих у вигляді функціональної або структурної схеми. При цьому, як відзначалося в п. 1.2.1, моделі в даному пакеті можуть зображати процеси будь-якої фізичної природи. Один і той же блок пакета *Simulink*, у залежності від специфіки розв'язуваної задачі, може являти собою функціональну модель різних пристроїв. Сполучні лінії між блоками *Simulink* показують передачу інформації або сигналу від одного елемента моделі до іншого, тому вони мають напрямок. У пакеті *SPS* блоки являють собою моделі різних елементів електричного кола або електромеханічних пристроїв, а сполучні лінії – це моделі електричних провідників, тому вони не мають напрямку (починаючи з версії MATLAB 6.5.1). При цьому зовнішній вигляд блока близький до зображення елемента на принциповій схемі.

Таким чином, пакет *SPS* дає змогу створити віртуальну модель, яка за своїми властивостями наближена до реального фізичного об'єкта.

З'єднання в одній моделі блоків двох пакетів допускається при врахуванні деяких обмежень, а саме:

- 1) порти *Simulink*-блоків, марковані знаком «>», призначені для підключення тільки *Simulink*-сигналу (тобто сигналу від іншого блока пакета *Simulink*); при цьому сигнал є безрозмірним і має певний напрямок, позначений стрілкою, а порт може бути або вхідним, або вихідним;
- 2) порти *SPS*-блоків, марковані знаком «□», є затискачами для включення у віртуальне електричне коло і можуть бути приєднані тільки до відповідних портів інших *SPS*-блоків або до лінії між *SPS*-блоками; при цьому моделюється електричне з'єднання елементів;
- 3) деякі *SPS*-блоки мають два види портів – для *SPS*-сигналу і для *Simulink*-сигналу, з відповідним маркуванням. Такі блоки дають змогу об'єднувати елементи двох пакетів в одну загальну модель.

Доступ до пакета *SPS* можна одержати через браузер бібліотек *Simulink*, а можна відкрити його окремо, записавши в командному вікні після курсору команду *powerlib*. Крім того, для багатьох *SPS*-блоків існує *Simulink*-аналог, тобто модель, яка виконує аналогічну функцію, але складена з блоків пакета *Simulink*. Ці моделі не мають власного вікна налаштування і відкриваються як звичайні підсистеми, завдяки чому їхній вміст цілком доступний для огляду і модифікації. Вони знаходяться у файлі *powerlib_models.mdl*, що розташований у папці *toolbox/physmod/powersys/powersys*. Відкрити даний файл можна безпосередньо через командне вікно, записавши в ньому ім'я *powerlib_models*.

У даному розділі розглянемо особливості моделювання різних електричних кіл, стандартні блоки для яких знаходяться переважно в бібліотеках *Electrical Sources*, *Elements* і *Measurements* пакета *SPS*. Додатково як керуючі елементи використовуються блоки пакета *Simulink*, розглянуті в розділі 1.

2.2 Моделювання електричних кіл однофазного синусоїдного струму

Моделі основних елементів, призначених для моделювання електричних кіл, розглянемо на прикладі кола однофазного синусоїдного струму, схема якого наведена на рис. 2.1, а, а модель – на рис. 2.1, б. Значення параметрів кола показані на схемі.

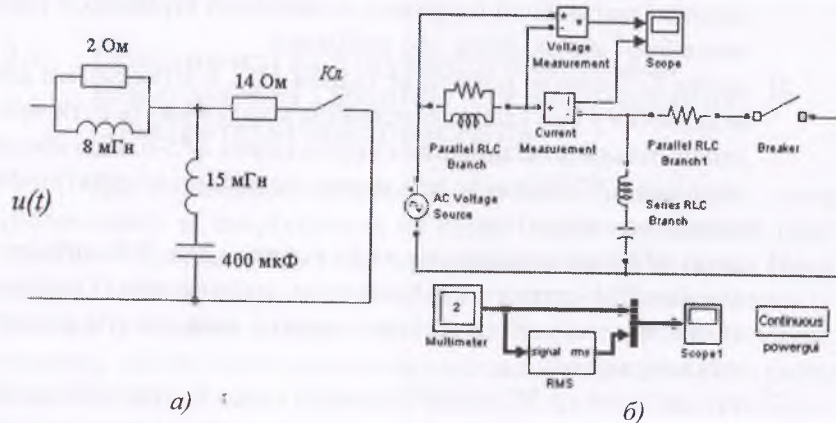


Рисунок 2.1 – Схема електричного кола синусоїдного струму (а) та її модель (б)

2.2.1 Моделі джерел синусоїдних електричних сигналів

Моделі джерел знаходяться в бібліотеці *Electrical Sources* пакета *SPS*. У даному параграфі розглянемо два з них – джерело синусоїдної напруги *AC Voltage Source* (рис. 2.1, б) і синусоїдного струму *AC Current Source*. Ці блоки генерують віртуальний електричний сигнал (напругу або струм) відповідно до виразу

$$a = A \cdot \sin(\omega \cdot t + \varphi),$$

де a – миттєве значення сигналу; A – амплітуда (параметр *Peak Amplitude* у вікні налаштування блока), φ – початкова фаза в градусах (параметр *Phase*), ω – кутова частота, рад/с (параметр *Frequency* – частота f , Гц, що пов'язана з кутовою частотою залежністю $\omega = 2\pi f$). Параметр *Sample time* (час вибірки) являє собою значення періоду дискретизації, для неперервних систем він дорівнює

нулю. Стрілка на піктограмі блока *AC Current Source* вказує умовно додатний напрямок струму через джерело.

Обидва блоки є моделями ідеальних джерел: джерела напруги з нульовим внутрішнім опором та джерела струму – з нескінченним внутрішнім опором.

2.2.2 Моделі резистивно-індуктивно-ємнісних кіл

Ідеальні резистор R , котушка індуктивності L і конденсатор C , а також послідовне з'єднання цих елементів моделюються універсальним блоком *Series RLC Branch*, а їх паралельне з'єднання – блоком *Parallel RLC Branch* (бібліотека *Elements*), рис. 2.2.

У вікнах налаштування блоків задаються параметри елементів: *Resistance* – опір (Ом), *Inductance* – індуктивність (Гн), *Capacitance* – ємність (Ф). Щоб виключити будь-який з елементів R , L чи C з послідовного блока, досить установити параметр даного елемента в значення 0, 0 та *inf* (*infinity* – нескінченність) відповідно, тобто обнулити його опір. Для того, щоб виключити елемент із паралельного блока, треба розімкнути гілку з даним елементом (задати нескінченний опір), тобто установити параметр R , L чи C у значення *inf*, *inf* та 0 відповідно. Відсутній елемент не відображається на піктограмі блока.

У версіях MATLAB, починаючи з 7.0.4, у вікнах налаштування обох блоків присутній додатковий параметр – *Branch type*. Він дає можливість вибрати за допомогою списку, що випадає, будь-яке сполучення елементів *RLC*-кола. Таким чином, немає необхідності у спеціальному налаштуванні параметра відсутнього елемента.

У тому випадку, якщо замість опору, індуктивності та ємності задані відповідні потужності елементів, треба використовувати блоки *Series RLC Load* і *Parallel RLC Load*. Вони також являють собою послідовне і паралельне з'єднання елементів R , L , C , але параметри кола задаються через діюче значення номінальної напруги (V_r), номінальну частоту (f_r), активну (P), індуктивну (Q_L) і ємнісну (Q_C) потужності. Для виключення з кола будь-якого елемента треба встановити його потужність у нульове значення.

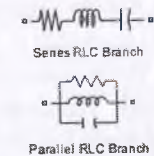


Рисунок 2.2 – Блок *Parallel RLC Branch*

2.2.3 Модель вимикача

Автоматичний вимикач чи рубильник у колі синусоїдного струму моделюється за допомогою блока *Breaker* (бібліотека *Elements*), керування яким здійснюється за сигналом або від зовнішнього пристрою (*Simulink*-сигнал), або від внутрішнього керуючого таймера.

У режимі зовнішнього керування блок має додатковий керуючий вхід *c* (рис. 2.3), сигнал на якому може дорівнювати 0 або 1 (0 – ключ розмикається, 1 – замикається). Момент часу, в який відбувається переключення, визначається часом спрацювання керуючого пристрою на вході *c*. У ролі такого пристрою можуть бути використані блоки *Step*, *Timer*, *Signal Generator* пакета *Simulink*. Початковий стан ключа визначається параметром *Initial state* блока *Breaker*: 0 – розімкнутий, 1 – замкнутий.

Режим внутрішнього керування здійснюється за допомогою налаштування параметрів блока *Breaker*. У цьому режимі піктограма блока має вигляд, як у моделі на рис. 2.1, б, тобто без керуючого входу. Вікно налаштування блока для даного випадку показано на рис. 2.4. У цьому вікні знятий прапорець *External control of switching times*, після чого з'явився додатковий параметр *Switching times*, призначений для введення часу спрацювання ключа. Якщо передбачається кілька спрацювань, то в даному рядку треба ввести ряд значень часу, розділених пробілом, у квадратних дужках. У цьому випадку в кожен заданий момент часу ключ переходить у стан, протилежний попередньому (тобто якщо був розімкнутий – то замикається, і навпаки).

Варто підкреслити, що розмикання ключа відбувається тільки

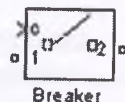


Рисунок 2.3 – Блок *Breaker*

Breaker resistance R_{on} (Ohm):	0.001
Initial state { 0 for 'open', 1 for 'closed' }:	0
Snubber resistance R_s (Ohms):	inf
Snubber capacitance C_s (F):	0
Switching times (s):	0.1
<input type="checkbox"/> External control of switching times	
Measurements:	None

Рисунок 2.4 – Вікно налаштування параметрів блока *Breaker*

в момент переходу струму через нуль. Тобто спочатку надходить сигнал на відключення (від зовнішнього пристрою чи від внутрішнього таймера), потім перевіряється значення струму через ключ, і лише при досягненні струмом найближчого нульового значення ключ розмикається. Ця умова реалізована з метою не допустити виникнення електричної дуги при розриві кола. Тому блок *Breaker* не підходить для розмикання кіл постійного струму. У таких колах рекомендується використовувати блок *Ideal Switch* (див. п. 3.1.1).

У замкнутому стані *Breaker* має невеликий опір R_{on} , що встановлюється в його вікні налаштування. У розімкнутому стані R_{on} дорівнює нескінченності. Параметри R_s і C_s призначені для введення до складу моделі послідовної демпфуючої *RC*-ланки, чи снаббера (snubber), паралельно вимикачу. Снаббер застосовується, якщо блок *Breaker* потрібно з'єднати послідовно з джерелом струму чи котушкою індуктивності (докладніше див. п. 2.8). У більшості випадків достатньо застосувати чисто резистивний снаббер ($C_s = inf$) з великим значенням активного опору R_s (порядку 1 МОм). При наявності конденсатора C_s снаббер виконує також функцію захисту від перенапруг. Щоб його повністю відключити, треба встановити $R_s = inf$ чи $C_s = 0$.

2.2.4 Модель трансформатора

Лінійний трансформатор *Linear Transformer* (бібліотека *Elements*) моделюється класичною Т-подібною схемою заміщення (рис. 2.5). Характеристика намагнічування сердечника являє собою лінійну залежність. За замовчуванням трансформатор має дві вторинні обмотки, одну з яких можна відключити, знявши прапорець *Three windings transformer* у вікні налаштування (рис. 2.6). Параметри обмоток – напруга V_{rms} , опір R та індуктивність L – задаються в полях *Winding 1(2,3) parameters* (тут і далі *RMS* означає «root mean square», тобто середньоквадратичне, або діюче, значення напруги). Опори й індуктивності можуть бути задані одним із двох способів: в одиницях системи СІ або відносних одиницях *p.u.* (*per unit*). Спосіб задання вибирається в графі *Units* вікна налаштування. Для

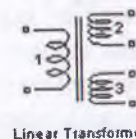


Рисунок 2.5 – Блок *Linear Transformer*

переведення параметрів з однієї системи в іншу програма використовує базові значення опору R_{base} та індуктивності L_{base} , які обчислюються за допомогою заданих номінальних потужності P_n , напруг обмоток V_n і частоти f_n :

$$R_{base} = \frac{V_n^2}{P_n}; \quad L_{base} = \frac{R_{base}}{2\pi f_n}.$$

Для кожної обмотки вирази опору і індуктивності у відносних одиницях мають вигляд:

$$R_{p.u.} = \frac{R_{\Omega}}{R_{base}}; \quad L_{p.u.} = \frac{L_H}{L_{base}},$$

де R_{Ω} і L_H – опір в Омах (Ω) та індуктивність у Генрі (H) відповідно.

Для розрахунку параметрів гілки намагнічування R_m і L_m у відносних одиницях використовуються базові значення R_{base} і L_{base} первинної обмотки. За замовчуванням $R_m = L_m = 500 \text{ p.u.}$ (поле *Magnetization resistance and reactance* на рис. 2.6), що відповідає струму намагнічування 0,2 % від номінального.

Для моделювання ідеального трансформатора (без втрат) потрібно встановити опори $R_{p.u.}$ й індуктивності $L_{p.u.}$ обмоток у 0, а параметри гілки намагнічування R_m і L_m – у нескінченність (*inf*).

2.2.5 Моделі вимірювальних приладів

2.2.5.1 Вимірники миттєвих значень струмів і напруг

Для виміру миттєвих значень струмів і напруг в електричному колі використовуються блоки *Current Measurement* і *Voltage Measurement* відповідно (бібліотека *Measurements*), рис. 2.7. Вимірник струму включається в коло послідовно, вимірник напруги – паралельно (див. рис. 2.1, б, 2.10). Як зазначалося в п. 2.1, для включення у віртуальне коло використовуються *SPS-порти* блоків з маркуванням □. До виходів *i* та *v* можуть бути підключені тільки елементи пакета *Simulink* (про що свідчить значок >). Таким чином, обидва блоки – *Current Measurement* і *Voltage Measurement* –

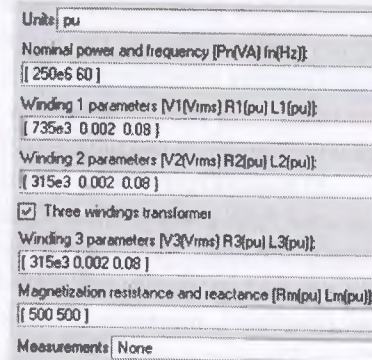


Рисунок 2.6 – Вікно налаштування блока *Linear Transformer*

являють собою свого роду інтерфейс для об'єднання в одну модель блоків пакетів *Sim Power Systems* і *Simulink*, тобто для перетворення віртуальних сигналів (струмів і напруг) у функціональні (безрозмірні) сигнали.

Існує інший спосіб вимірювання електричних величин – застосування універсального вимірника *Multimeter*. Особливість цього блока полягає в тому, що він не має входних портів, тобто не потребує підключення до кола, тому може бути розміщений у будь-якому місці вікна моделі (див. рис. 2.1, б). Вимірювані сигнали попередньо вибираються в графі *Measurements* вікон налаштування блоків, які моделюють різні елементи кола, зокрема *Series (Parallel) RLC Branch*, *Breaker* (рис. 2.4), *Linear Transformer* (див. рис. 2.6) та ін. Установка цього параметра в значення *Branch voltage*, *Branch current* чи *Branch voltage and current* еквівалентна підключенню внутрішнього вимірника струму або/і напруги всередині даного блока.

Після цього обрані сигнали відображаються в лівому полі *Available Measurements* вікна налаштування блока *Multimeter* (рис. 2.8). Далі, виділивши потрібні сигнали мишею, необхідно за допомогою кнопки >> перенести їх у праве поле *Selected Measurements* і натиснути *Close*.

На піктограмі блока з'явиться число, що відповідає кількості вимірюваних сигналів. У моделі на рис. 2.1, б за допомогою даного

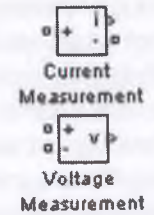


Рисунок 2.7 – Блоки *Current Measurement* і *Voltage Measurement*

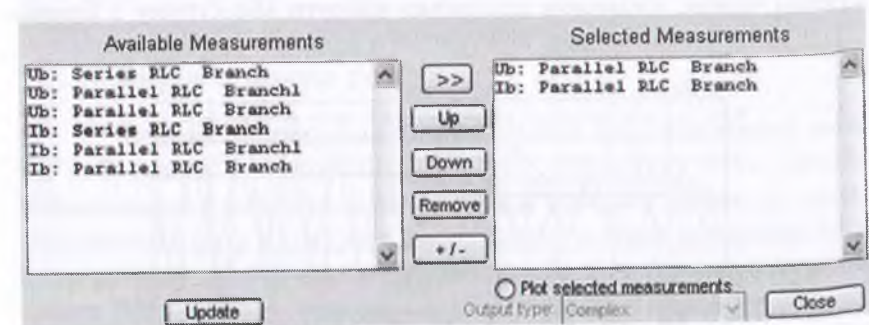


Рисунок 2.8 – Вікно налаштування блока *Multimeter*

блока вимірюються напруга і струм паралельного кола *Parallel RLC Branch* (сигнали U_b та I_b у правому вікні на рис. 2.8).

Якщо вимірюваних сигналів декілька, то можна використовувати блок *Demux* для поділу векторного виходу вимірника на окремі сигнали, кожний з яких подати на окремий вхід осцилографа *Scope*. Можна також зовсім обійтися без блока *Scope*. Для цього у вимірнику *Multimeter* потрібно встановити прапорець *Plot selected measurements* (див. рис. 2.8). При запуску моделювання автоматично відкриється додаткове вікно, у якому будуть відображені графіки для обраних сигналів. Це вікно містить меню і панель інструментів, отже, графіки можуть бути відредаговані, а потім скопійовані в будь-який графічний чи текстовий редактор (меню *Edit* → *Copy Figure*, потім перейти в потрібний редактор і виконати команду «Вставка»).

2.2.5.2 Вимірник діючого значення сигналу

Якщо потрібно виміряти діюче значення напруги чи струму, то вихід відповідного вимірника треба підключити до блока *RMS* (бібліотека *Extra Library/Measurements*), рис. 2.1, б. Він розраховує середньоквадратичне (діюче) значення вхідного сигналу $f(t)$ за формулою

$$RMS = \sqrt{\frac{1}{T} \int_{t-T}^t f(t)^2 dt}$$

де $T = 1/f$ – період сигналу; f – частота основної гармоніки (*fundamental frequency*, задається у вікні налаштування блока *RMS*). Таким чином, з'єднання вимірника напруги або струму з блоком *RMS* являє собою модель віртуального вольтметра або амперметра.

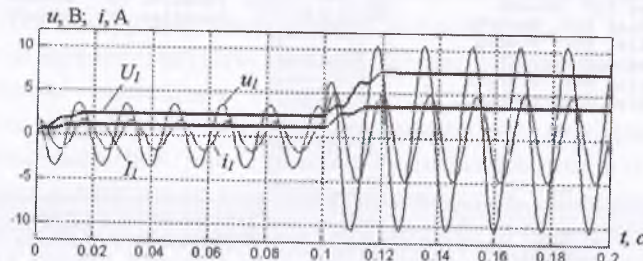


Рисунок 2.9. – Осцилограми струмів і напруг в моделі на рис. 2.1, б

Графіки перехідного процесу в досліджуваному колі (див. рис. 2.1) зображені на рис. 2.9: u_1 , i_1 – миттєві напруга і струм паралельної *RL*-ділянки кола; U_1 , I_1 – їхні діючі значення, отримані за допомогою блока *RMS*. Видно, що параметри всіх сигналів змінюються в момент часу 0,1 с, коли замикається ключ і підключається активний опір паралельно *RC*-ланці. У результаті повний опір кола зменшується, тому зростають струм і напруга на паралельній *RL*-ланці.

2.2.5.3 Вимірник потужності

Для виміру активної і реактивної потужності кола застосовується блок *Active & Reactive Power* (бібліотека *Extra Library/Measurements*), на входи V та I якого подаються сигнали від вимірників напруги і струму відповідно (рис. 2.10). Єдиним параметром блока є частота основної гармоніки *fundamental frequency*. Потужності розраховуються за формулами

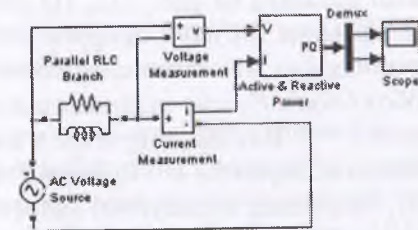


Рисунок 2.10 – Модель електричного кола з вимірником потужності

$$P = \frac{1}{T} \int_{t-T}^t (V(\omega t) \times I(\omega t)) dt; \quad Q = \frac{1}{T} \int_{t-T}^t \left(V(\omega t) \times I\left(\omega t - \frac{\pi}{2}\right) \right) dt.$$

Для розподілу активної і реактивної потужностей на окремі сигнали рекомендується використовувати блок *Demux* на виході вимірника.

2.2.5.4 Вимірник різниці фаз

Модель приладу для виміру різниці фаз (фазометра) можна побудувати на основі блока *Fourier* (бібліотека *Extra Library/Measurements*), рис. 2.11, 2.12. Цей блок вимірює амплітуду (вихід *Magnitude*) і фазу в градусах (вихід *Angle*) заданої гармоніки вхідного сигналу, номер якої n вводиться в параметрах налаштування блока. Для основної гармоніки $n = 1$. На рис. 2.11 фазометр реалізований у вигляді підсистеми з іменем *Phasometer*, внутрішня модель

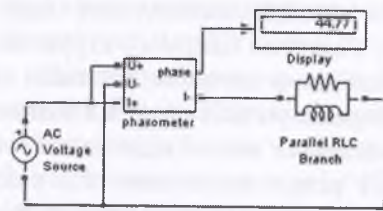


Рисунок 2.11 – Модель електричного кола з вимірником різниці фаз *Phasometer*

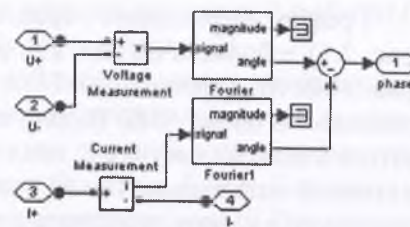


Рисунок 2.12 – Внутрішня модель підсистеми *Phasometer*

якої показана на рис. 2.12. На входи підсистеми надходять миттєві значення струму і напруги від вимірників, з виходу знімається різниця фаз між цими сигналами. Параметр *fundamental frequency* обох блоків *Fourier* задається рівним частоті джерела, номер гармоніки $n = 1$. Виходи *Magnitude* у даній моделі не використовуються. Задаючи параметр $n = 0$, можна виміряти постійну складову сигналу, наприклад, пульсуючої напруги на виході випрямляча. При цьому необхідно пам'ятати, що при двопульсовому випрямленні частота основної гармоніки дорівнює подвоєній частоті джерела живлення.

2.2.5.5 Вимірник частоти

У бібліотеці елементів *SPS* відсутній віртуальний прилад для виміру частоти, але його можна скласти самостійно з блоків пакета *Simulink* (див. п. 1.4.2.2, рис. 1.36). При включенні віртуального частотоміра в модель електричного кола замість сигналу від блока *Signal Generator* треба використовувати електричний сигнал (отриманий за допомогою вимірника струму або напруги), частоту якого потрібно виміряти.

2.2.5.6 Інтерфейс *Steady-State Voltages and Currents* блока *Powergui*

Крім описаних вище блоків, у пакеті *SPS* існує додаткова можливість виміру амплітудних і діючих значень напруг та струмів у моделі, а також їхніх фаз. Для цього використовується блок *Powergui* (*Power graphical user interface*, бібліотека *Sim Power Systems*), рис. 2.1, б. В останніх версіях MATLAB даний блок з'являється у вікні будь-якої *SPS*-моделі автоматично при запуску

моделювання, тому немає необхідності переносити його з бібліотеки елементів. Основне його призначення – забезпечити графічний інтерфейс для всебічного аналізу *SPS*-моделей. Цей блок не має вхідних і вихідних портів, тому може бути розміщений у будь-якому місці вікна моделі. Вікно його налаштування містить ряд кнопок. Для виміру амплітуди і фази електричних сигналів використовується кнопка *Steady-State Voltages and Currents*. При її натисканні відкривається вікно, у якому можуть відображатися виміряні значення наступних сигналів (рис. 2.13):

States – змінні стану моделі (струми котушок і напруги конденсаторів);

Measurements – сигнали, обрані як вимірювані у вікнах налаштування блоків (поле *Measurements*);

Sources – сигнали джерел;

Nonlinear Elements – струми і напруги нелінійних елементів.

Всі перераховані види вимірів вибираються праворуч у графі *Display*. У графі *Units* указується, які саме значення сигналів повинні бути виміряні – амплітудні (*Peak Values*) або діючі (*RMS Values*). Варто підкреслити, що для відображення сигналів в опціях *Measurements* і *Sources* даного вікна ці сигнали повинні бути обрані як вимірювані у відповідних блоках. Наприклад, у вікні налаштування блока *AC Voltage Source* параметр *Measurements* має бути встановлений у значення *Voltage*. Інші можливості блока *Powergui* будуть розглянуті в наступних розділах даного посібника.

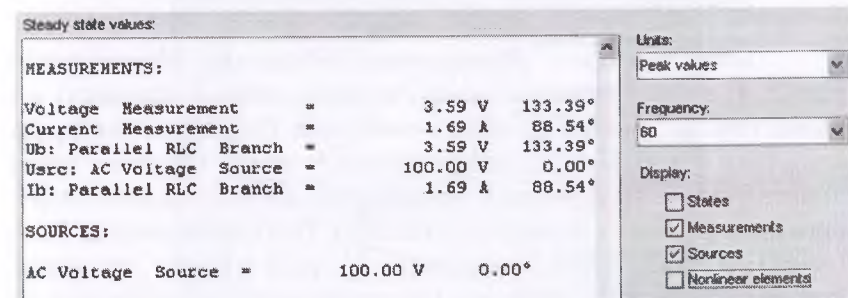


Рисунок 2.13 – Вікно інтерфейсу *Steady-State Voltages and Currents* блока *Powergui*

2.2.6 Одержання характеристик електричного кола

Завдяки узгодженій взаємодії пакетів середовища MATLAB розроблювач має можливість не тільки виміряти значення віртуальних сигналів у моделі електричного кола, але також побудувати його часові та частотні характеристики, одержати передатну функцію, значення коефіцієнтів демпфування та ін. Для цього використовується пакет *Control System Toolbox*. Його застосування для дослідження функціональної *Simulink*-моделі розглядалося в п. 1.3.2. У пакеті *Sim Power Systems* також існують кілька інструментів, що дають змогу досліджувати віртуальну модель методами теорії автоматичного керування з залученням можливостей пакета *Control System Toolbox*.

Як приклад розглядається електричне коло, схема і значення параметрів якого наведені на рис. 2.1, а. Модель кола для дослідження частотних характеристик показана на рис. 2.14. Ключ у гілці з активним опором вважається замкнутим, тому в даній моделі він відсутній. Замість нього встановлений ключ на вході кола, який знаходиться в розімкнутому стані.

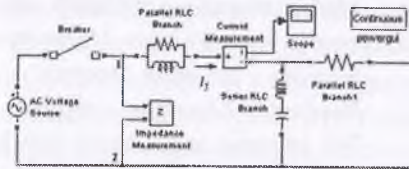


Рисунок 2.14 – Модель електричного кола для дослідження частотних характеристик

2.2.6.1 Інтерфейс Impedance vs Frequency Measurement блока Powergui

Блок *Impedance Measurement* (бібліотека *Measurements*), рис. 2.14, призначений для виміру повного опору (імпедансу) між двома точками лінійного електричного кола. Результати виміру можуть бути отримані за допомогою блока *Powergui* і подані у вигляді частотних характеристик – залежностей модуля повного опору і його фази від частоти джерела живлення. Внутрішня модель блока *Impedance Measurement* складається з джерела струму I_Z , підключеного між двома входами блока, і паралельного йому вимірника напруги V_Z . Імпеданс кола розраховується як передатна функція $H_Z(s)$:

$$H_Z(s) = \frac{V_Z(s)}{I_Z(s)}$$

Блок підключається до двох точок моделі, між якими потрібно виміряти імпеданс (точки 1 і 2 на рис. 2.14). Якщо коло замкнуте, то імпеданс між цими точками дорівнює нулю, оскільки ідеальне джерело має нульовий опір. Тому, якщо потрібно виміряти імпеданс усього кола, треба розімкнути коло біля джерела, видаливши сполучну лінію, або підключити на вході блок *Breaker* у розімкнутому стані (див. рис. 2.14). Процес моделювання запускати не потрібно. Далі відкрити блок *Powergui*, вибрати пункт *Impedance vs Frequency Measurement* і у вікні, що відкрилося, натиснути кнопку *Display*. У результаті будуть отримані два графіки – частотні залежності модуля Z і фази ϕ повного опору кола (рис. 2.15). У правій частині вікна знаходиться панель налаштувань, за допомогою якої можна задати шкалу графіків (логарифмічну або лінійну), вибрати частотний діапазон і крок розрахунку (*Range*), відобразити координатну сітку (*Grid*).

Отримані залежності можна використовувати для аналізу поведінки електричного кола на всьому діапазоні частот і, зокрема, для визначення резонансної частоти $f_{\text{рез}}$. Так, на рис. 2.15 видно, що при $f \rightarrow 0$ і $f \rightarrow \infty$ опір досліджуваного кола носить чисто активний характер (значення фази прямує до нуля), причому $Z = R_2 = 14$ Ом при $f = 0$ і $Z = R_1 + R_2 = 16$ Ом при $f \rightarrow \infty$. З графіків видно також, що в діапазоні частот від 10 до 100 Гц у колі відбувається резонанс напруг (опір досягає мінімуму, фаза проходить через нуль).

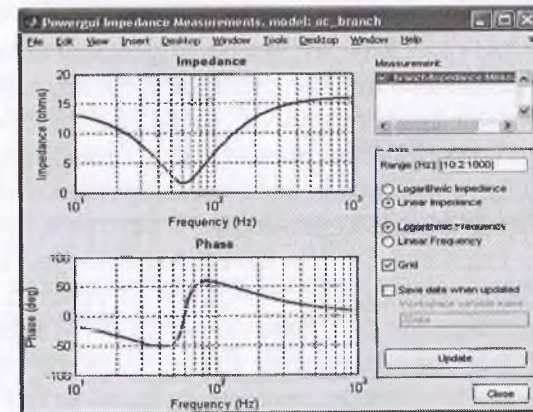


Рисунок 2.15 – Вікно інтерфейсу *Impedance Measurement*

Для знаходження точного значення резонансної частоти потрібно вибрати в меню *Tools* → *Data Cursor*. Після цього, помістивши покажчик миші в будь-яку точку на графіку, у додатковому віконці, що з'явилося, можна побачити значення координат даної точки по осях *X* та *Y*. У даному випадку $f_{\text{рез}} = 60$ Гц, $Z_{\text{рез}} = 1,46$ Ом.

Якщо потрібно з'єднати вимірник імпедансу послідовно з коутшкою індуктивності, джерелом струму або яким-небудь нелінійним елементом, то паралельно блоку *Impedance Measurement* треба включити резистор з великим опором (докладніше див. п. 2.8).

2.2.6.2 Інтерфейс *LTI Viewer* блока *Powergui*

Додаткові можливості для дослідження лінійного електричного кола методами теорії керування надає спеціальний інтерфейс *LTI Viewer*, що активізується за допомогою блока *Powergui* (пункт *Use LTI Viewer*). Вимірник імпедансу тепер можна відключити. Після запуску інтерфейсу *LTI Viewer* відкривається вікно, в якому потрібно вибрати зі списків, який сигнал у колі буде вважатися вхідним, а який – вихідним. Як вхідний сигнал зазвичай використовується напруга або струм відповідного джерела, а вихідний сигнал – це напруга або струм одного з вимірників (тому в колі має бути хоча б один вимірник). У досліджуваній моделі на рис. 2.14 за вихідний сигнал приймемо струм I_1 , вимірюваний блоком *Current Measurement*. Вибір підтверджується натисканням кнопки *Open new LTI Viewer*, після чого відкривається вікно *LTI Viewer* з перехідною характеристикою (рис. 2.16, а), яка подає реакцію кола на одиничний стрибок задаючого впливу. Іншими словами, характеристика на рис. 2.16, а – це графік струму I_1 , якщо замість синусоїдної напруги на вході кола стрибком прикладена постійна напруга 1 В.

Можна побудувати також інші характеристики електричного кола. Для цього в меню *Edit* вікна *LTI Viewer* треба натиснути *Plot Configurations* і далі праворуч на панелі *Response type* з випадючого списку вибрати бажану характеристику. Як приклад, розглянемо амплітудно-частотну (АЧХ) і фазочастотну (ФЧХ) характеристики (рис. 2.16, б). Для їхньої побудови в графі *Response type* був обраний пункт *Bode*. Подвійним натисненням миші на осях координат відкривається меню налаштування (*Property Editor*), в якому можна вибрати одиниці виміру усіх величин – амплітуди (децибели або

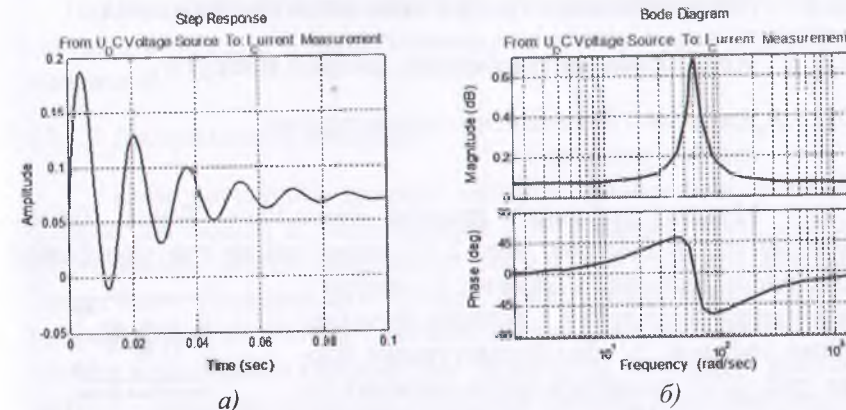


Рисунок 2.16 – Характеристики електричного кола: перехідна (а), частотні (б)

абсолютні одиниці), фази (градуси або радіани) і частоти (рад/с або Гц). Амплітудно-частотна характеристика показує, як змінюється амплітуда струму I_1 на всьому діапазоні частот, якщо амплітуда синусоїдної напруги на вході 1 В, а ФЧХ показує зміну різниці фаз між вихідним і вхідним сигналами. При резонансі напруг струм у колі максимальний, про що свідчить сплеск на АЧХ, а різниця фаз дорівнює нулю, оскільки струм збігається по фазі з вхідною напругою. Резонансна частота $f_{\text{рез}} = 60$ Гц відповідає знайденій раніше за графіком повного імпедансу кола.

За допомогою даного інтерфейсу можна одержати передатну функцію $H(s)$ електричного кола. Для цього потрібно в меню вікна *LTI Viewer* вибрати команди *File* → *Export*, у вікні, що відкрилося, у графі *Export...* ввести ім'я, під яким передатна функція буде збережена в робочій області (наприклад, *model*) і натиснути кнопку *Export to Workspace*. Далі в командному вікні MATLAB ввести команду *tf(model)* і натиснути *Enter*. Після цього з'явиться передатна функція досліджуваного електричного кола. Для аналізу систем у формі передатних функцій застосовується пакет *Control System Toolbox* середовища MATLAB, розгляд якого виходить за рамки даного посібника.

2.3 Моделювання трифазних електричних кіл

2.3.1 Моделювання трифазних джерел напруги

2.3.1.1 Джерело з постійними параметрами вихідної напруги

Модель трифазного джерела *Three-Phase Source* (бібліотека *Electrical Sources*), рис. 2.17, являє собою три однофазних ідеальних джерела, з'єднаних у «зірку» і включених послідовно з активно-індуктивними опорами. У вікні налаштування блока (рис. 2.18) задаються діюче значення лінійної напруги (*Phase-to-phase rms voltage*), фазовий кут фази А в градусах (*Phase angle of phase A*), частота напруги (*Frequency*), а також вид з'єднання фаз (*Internal connection*) – «зірка» (Y), «зірка із заземленням» (Yg), «зірка з нульовим проводом» (Yn). В останньому випадку в блоці з'являється додатковий порт для підключення нульового проводу.

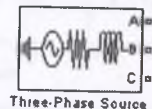


Рисунок 2.17 – Блок *Three-Phase Source*

Внутрішні параметри джерела можна задати одним із двох способів. Перший спосіб – зняти прапорець *Specify impedance using short-circuit level* і безпосередньо ввести в поля, що з'явилися, значення опору й індуктивності. Другий спосіб – при встановленому прапорці (рис. 2.18) задати параметри кола в режимі короткого замикання: індуктивну потужність Q_{sc} , яка виділяється в колі в цьому режимі (поле *3-phase short-circuit level*), і базову напругу V_{base} (поле *Base voltage*). При цьому внутрішня індуктивність джерела розраховується за формулою

$$L = \frac{V_{base}^2}{Q_{sc}} \cdot \frac{1}{2\pi f}$$

Рисунок 2.18 – Вікно налаштування блока *Three-Phase Source*

Phase-to-phase rms voltage (V):	25e3
Phase angle of phase A (degrees):	0
Frequency (Hz):	60
Internal connection:	Yg
<input checked="" type="checkbox"/> Specify impedance using short-circuit level	
3-phase short-circuit level at base voltage (VA):	100e6
Base voltage (V rms ph-ph):	25e3
X/R ratio:	7

Параметр *X/R ratio* визначає співвідношення між індуктивним і активним опором. З цього співвідношення програма розраховує значення R .

2.3.1.2 Програмоване джерело

Крім звичайного джерела напруги, у бібліотеці елементів є також дві моделі зі змінюваними в часі параметрами: *3-phase Programmable Source* (бібліотека *Extra Library/Control Blocks*) і *Three-Phase Programmable Voltage Source* (бібліотека *Electrical Sources*), рис. 2.19. Це програмовані джерела, які дають змогу задати зміну в часі одного з параметрів – амплітуди, початкової фази або частоти – основної гармоніки вихідного сигналу. Існує можливість також сформувати додатково будь-які дві гармоніки і накласти на основну. Різниця між двома блоками полягає в тому, що на виході блока *3-phase Programmable Source* знаходиться вектор, який містить три безрозмірних *Simulink*-сигнали, зсунутих за фазою на 120° . Тому він може бути використаний як джерело керуючого впливу в моделях систем керування, виконаних на базі пакета *Simulink* (див. п. 4, рис. 4.44). Блок *Three-Phase Programmable Voltage Source* аналогічний попередньому за принципом роботи і параметрами вікна налаштування, але має три виходи, з яких знімається віртуальна трифазна напруга, тобто *SPS*-сигнал. Отже, цей блок є моделлю джерела напруги і може бути використаний у моделях електричних кіл пакета *Sim Power System*. Порт N – це вивід нульової точки джерела, з'єднаного за схемою «зірка».

На рис. 2.20, а показане вікно налаштування блока *Three-Phase Programmable Voltage Source*. У верхньому полі задаються параметри трифазної системи: діюче значення лінійної напруги, фазовий кут (у градусах) і частота фази А при прямому чергуванні фаз (*Positive-sequence*). У полі *Time variation of* вибирається параметр, який має змінюватися в процесі моделювання: амплітуда, початкова фаза або частота. До кожного з зазначених параметрів

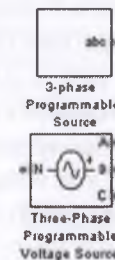


Рисунок 2.19 – Блоки *3-phase Programmable Source* і *Three-Phase Programmable Voltage Source*

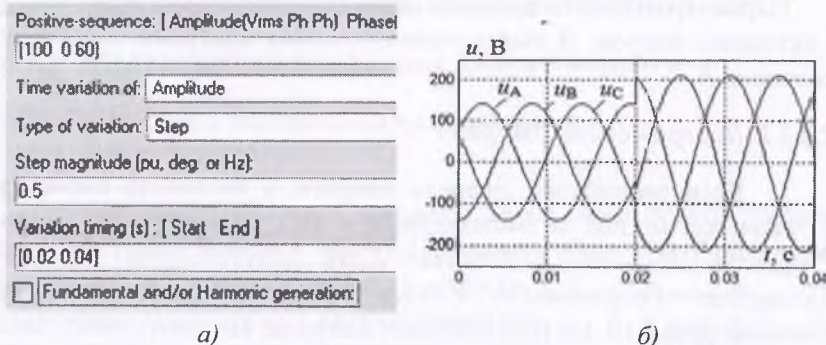


Рисунок 2.20 – Вікно налаштування (а) та осцилограми (б) блока *Three-Phase Programmable Voltage Source*

може бути застосований один з типів зміни (поле *Type of variation*): зміна стрибком (*Step*), зміна за лінійною залежністю (*Ramp*) або модуляція (*Modulation*). Коли тип зміни обраний, з'являються додаткові опції налаштування, характерні для даного типу. Так, для типу *Step* (рис. 2.20, а) задається розмір стрибка *Step magnitude* змінюваного параметра, для типу *Ramp* – швидкість його зміни *Rate of change*, для типу *Modulation* – амплітуда і частота модулюючого сигналу. Поле *Variation timing* містить два значення часу: час початку і час завершення заданої зміни. Для програмування двох додаткових гармонік треба встановити прапорець *Fundamental and/or Harmonic generation* і ввести в полях, що з'явилися, для кожної гармоніки її номер, амплітуду та фазовий кут, а також час початку і кінця дії цих гармонік.

На рис. 2.20, б показані осцилограми на виході даного блока, налаштованого відповідно до рис. 2.20, а. У момент часу 0,02 с амплітуда напруги стрибком зростає на $0,5U_m$ щодо початкового значення $U_m = 100\sqrt{2}$ (у полі *Positive-sequence* задане діюче значення напруги 100 В).

2.3.1.3 Джерело із зовнішнім заданням параметрів

Особливість, яка обмежує використання блока *Three-Phase Programmable Voltage Source*, полягає в тому, що всі зміни параметрів вихідної напруги задаються вручну у вікні налаштування блока і не можуть надходити в його модель за допомогою зовнішніх

джерел сигналу. Крім того, вікно налаштування не дозволяє запрограмувати одночасну зміну більшу, ніж одного параметра. Однак у деяких випадках, наприклад, при моделюванні частотного керування за законом $U/f = \text{const}$, потрібно змінювати амплітуду напруги пропорційно його частоті. При цьому значення частоти може розраховуватися системою керування і надходити на джерело з зовнішньої моделі. У такому разі виникає потреба складання моделі джерела самостійно (рис. 2.21).

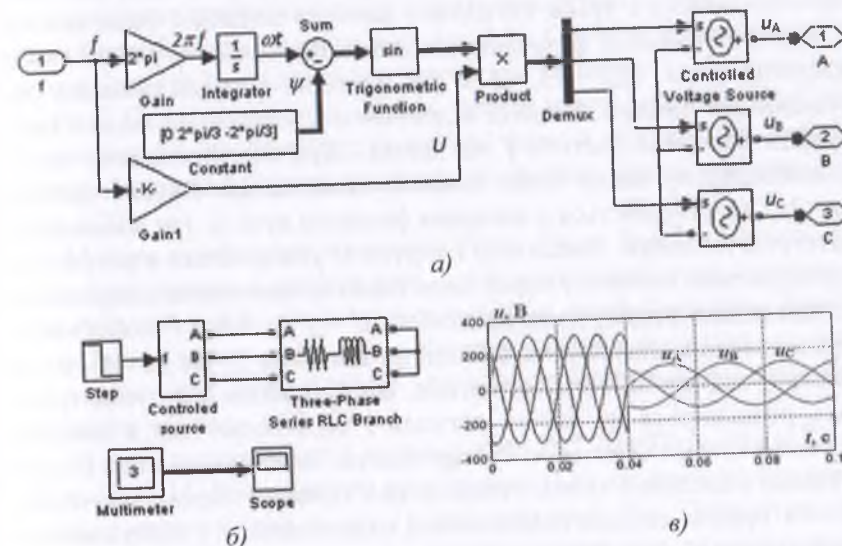


Рисунок 2.21 – Модель джерела *Controlled source* із зовнішнім заданням параметрів (а); модель системи з джерелом *Controlled source* (б); осцилограми напруги на навантаженні (в)

На рис. 2.21, а зображена модель керованого джерела напруги, яка реалізована у вигляді підсистеми *Controlled source* і включена в зовнішню модель трифазного кола з активно-індуктивним навантаженням (рис. 2.21, б).

У підсистемі *Controlled source* виконується розрахунок напруг трьох фаз відповідно до залежностей

$$u_A = U \cdot \sin(2\pi f \cdot t);$$

$$u_B = U \cdot \sin(2\pi f \cdot t - 2\pi/3);$$

$$u_c = U \cdot \sin(2\pi f \cdot t + 2\pi/3),$$

де частота f змінюється з 50 до 20 Гц у момент часу 0,04 с. Пропорційно частоті змінюється й амплітуда напруги U відповідно до залежності $U = f \cdot k$, де $k = U_{\text{ном}} / f_{\text{ном}}$ (тут $U_{\text{ном}} = 220\sqrt{2}$ В, $f_{\text{ном}} = 50$ Гц, $k = 6,2$ В/Гц).

Розглянемо докладніше роботу підсистеми на рис. 2.21, а. Синусоїдна функція обчислюється блоком *Trigonometric Function*, на вхід якого надходить значення аргументу у вигляді вектора, що складається з трьох сигналів – значень поточної фази кожної з напруг. Початкові фази задаються блоком *Constant*, у вікні якого введений вираз $[0,2 \cdot \pi/3, -2 \cdot \pi/3]$. Векторні сигнали показані потовщеними лініями. На вхід підсистеми з зовнішньої моделі надходить значення частоти f від блока *Step*, яке, проходячи через підсилювальну ланку *Gain*, помножується на 2π . Кутова частота $\omega = 2\pi f$ перетворюється в значення фазового кута $\omega \cdot t$ за допомогою інтегруючої ланки. Амплітуда напруги U утворюється в результаті проходження сигналу f через блок *Gain1* зі значенням коефіцієнта k (тим самим реалізується залежність $U = f \cdot k$). Блок *Product* виконує множення синусоїдної функції на значення U . На виході цього блока – векторний *Simulink*-сигнал, що відповідає напругам трьох фаз, розділяється на окремі сигнали і перетворюється в напругу трьома блоками *Controlled Voltage Source*. Затискачі «–» цих блоків з'єднані в спільний вузол, утворюючи з'єднання «зірка», а затискачі «+» призначені для підключення навантаження у віртуальному електричному колі. При необхідності спільна точка джерел може бути заземлена (блок *Ground*) або виведена з підсистеми в зовнішню модель (блок *Connection port*). Блок *Controlled Voltage Source* докладніше розглянутий у п. 2.5.1.

Осцилограми напруги на навантаженні, отримані за допомогою блока *Multimeter*, наведені на рис. 2.21, в. Видно, що в момент часу 0,04 с частота й амплітуда сигналу стрибком змінюють свої значення. Якщо потрібно реалізувати лінійно-наростаючий характер зміни, то замість блока *Step* на вході джерела треба використовувати блок *Ramp*. Також можна змінювати частоту й амплітуду незалежно один від одного. Для цього потрібно значення амплітуди U подавати на вхід блока *Product* від окремого джерела. Для зміни початкової фази в процесі моделювання треба підключити джерело *Step* або

Ramp замість блока *Constant*. При цьому необхідно враховувати, що це джерело сигналу має генерувати значення початкової фази для всіх трьох напруг, отже, у його налаштуваннях вводяться три значення. Крім того, різниця фаз між трьома напругами при будь-яких змінах має залишатися фіксованою, рівною $2\pi/3$.

2.3.2 Моделювання елементів трифазного кола

У бібліотеці *Elements* більшість елементів, призначених для моделювання трифазних кіл, у цілому аналогічні розглянутим раніше елементам однофазного кола (п. 2.2). Основна різниця полягає в тому, що трифазні елементи мають три затискачі з маркуваннями *A*, *B*, *C*. Отже, внутрішня модель блоків також містить у собі три компоненти – по одному для кожної фази, з однаковими параметрами. Зовнішній вигляд основних трифазних блоків показаний на рис. 2.22.

Модель трифазного вимикача *Three-Phase Breaker* містить три ключі, по одному в кожній фазі. Їхній початковий стан, однаковий для всіх трьох фаз, визначається вибором опції *Open* або *Closed* у полі *Initial status of breakers*. Спрацьовування вимикача задається окремо для кожної фази установкою відповідного прапорця *Switching of phase A (B, C)*.

Моделі послідовного і паралельного RLC-кіл *Three-Phase Series (Parallel) RLC Branch* нічим не відрізняються від аналогічних однофазних моделей (див. п. 2.2.2), крім кількості фаз. Параметри елементів R , L і C , що задаються у вікні налаштування, вважаються однаковими для трьох фаз. Якщо потрібні різні комбінації елементів чи значення параметрів (наприклад, при моделюванні несиметричного навантаження), то треба використовувати три однофазних блоки і настроювати кожний окремо. Блоки трифазного навантаження *Three-Phase Series (Parallel) RLC Load*, на відміну від однофазних аналогів, містять у вікні налаштування параметр *Configuration*, що визначає схему з'єднання фаз:

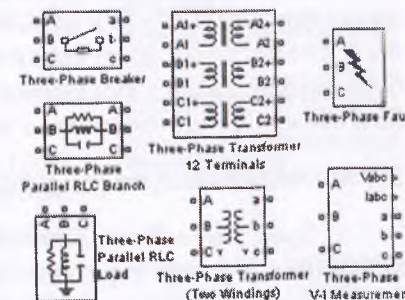


Рисунок 2.22 – Стандартні блоки для моделювання трифазних кіл

«зірка» (*Y floating*), «зірка з заземленням» (*Y grounded*), «зірка з нульовим проводом» (*Y neutral*) і «трикутник» (*Delta*). У полі *Nominal phase-to-phase voltage* задається діюче значення лінійної напруги.

У моделі трифазного трансформатора *Three-Phase Transformer* також існує можливість вибрати схему з'єднання для кожної з обмоток. Цей блок присутній у двох варіантах – *Two Windings* і *Three Windings*, які моделюють двообмотковий та триобмотковий трансформатори відповідно. Крім того, на відміну від однофазної лінійної моделі *Linear Transformer* (див. п. 2.2.4), ці блоки дають змогу моделювати нелінійність осердя. Для цього потрібно у вікні налаштування встановити прапорець *Saturable core*. У результаті з'являється додаткове поле *Saturation characteristic*, у якому задаються значення струму і магнітного потоку у відносних одиницях, що описують криву намагнічування. При необхідності можна також змоделювати гістерезис, для чого треба встановити прапорець *Simulate hysteresis*. У полі *Hysteresis data MAT file*, що з'являється, за замовчуванням вказане ім'я файлу *hysteresis.mat*, який містить необхідні дані. Щоб побачити петлю гістерезиса, задану в цьому файлі, потрібно скористатися інтерфейсом *Hysteresis Design tool* блока *Powergui*. У спеціальних полях інтерфейсу задані значення характерних точок кривої, а у вікні ліворуч показаний її зовнішній вигляд. Якщо ця крива не підходить, можна задати інші значення і натиснути кнопку *Update*, щоб відразу побачити зміни на графіку. Одержавши задовільний результат, потрібно натиснути кнопку *Save* і зберегти параметри кривої в новому *mat*-файлі.

Ще одна модель трифазного двообмоткового трансформатора – *Three-Phase Transformer 12 Terminals* – блок із дванадцятьма виводами (див. рис. 2.22). Він відрізняється від попередньої моделі тим, що в даному блоці доступні як початки, так і кінці обмоток. Може бути використаний у тих випадках, коли потрібно з'єднати обмотки не в «зірку» чи «трикутник», а якось інакше.

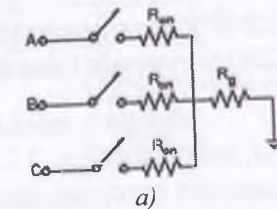
2.3.3 Моделювання кіл у режимі короткого замикання

Блок *Three-Phase Fault* (див. рис. 2.22) призначений для моделювання режиму короткого замикання в електричному колі. З його допомогою можна реалізувати замикання кількох фаз між собою, замикання фази на землю або комбінацію обох видів замикання. Його внутрішня модель показана на рис. 2.23, а.

Блок являє собою перемичку, яка за допомогою ключів з'єднує фази електричного кола між собою, створюючи тим самим аварійний режим. Перемичка також з'єднана із заземлюючим провідником, опір якого в нормальному режимі дуже великий (1 МОм), а в аварійному – дуже малий (порядка 0,001 Ом). Встановлюючи прапорці навпроти параметрів *Phase A (B, C) Fault* у вікні налаштування блока (рис. 2.23, б), можна вибрати, які саме фази будуть замикатися накоротко, тобто активізувати відповідні вимикачі. Для реалізації замикання фази (чи кількох фаз) на землю потрібно установити прапорець *Ground Fault*. При цьому з'являється додатковий параметр *Ground resistance Rg*, у якому можна задати опір заземлюючого провідника в аварійному режимі.

Керування блоком *Three-Phase Fault*, як і блоком *Breaker*, може бути реалізоване як від зовнішнього пристрою, так і від внутрішнього таймера. У першому випадку потрібно установити прапорець *External control of fault timing*, при цьому на піктограмі блока з'являється додатковий вхід для подачі керуючого сигналу (0 чи 1) від *Simulink*-блока, а у вікні налаштування – рядок *Initial status of fault*. Даний параметр визначає початковий стан кожної фази до подачі керуючого сигналу. Наприклад, вираз [0 0 1] означає, що спочатку у фазах А і В ключі були розімкнуті, а у фазі С – ключ замкнутий.

При внутрішньому керуванні (прапорець знятий, рис. 2.23, б) у вікні налаштування з'являються два параметри: *Transition status* і *Transition times*. Перший з них визначає стани обраних ключів (0 – розімкнутий, 1 – замкнутий), а другий – моменти часу, в які ці



а)

☒ Phase A Fault
☒ Phase B Fault
☒ Phase C Fault
 Fault resistances R_{on} (ohms):
 [0.001]
☒ Ground Fault
 Ground resistance R_g (ohms):
 [0.001]
☐ External control of fault timing:
 Transition status [1,0,1 ...]:
 [1 0 1]
 Transition times (s):
 [0.25 0.5 0.7]
 Snubbers resistance R_p (ohms):
 [1e6]
 Snubbers Capacitance C_p (Farad):
 [inf]
 Measurements: None

б)

Рисунок 2.23 – Внутрішня модель (а) і вікно налаштування (б) блока *Three-Phase Fault*

стани виникають. У вікні на рис. 2.23, б ці параметри встановлені в значення [1 0 1] та [0,25 0,5 0,7] відповідно. Це означає, що в момент часу 0,25 с ключі замикаються (1), у момент 0,5 с – розмикаються (0), а в 0,7 с – знову замикаються (1). Кількість елементів в обох рядках має бути однаковою.

Оскільки внутрішня модель даного блока містить у собі вимикачі, то до її складу, як і до складу блока *Breaker*, входить демпфувальна *RC*-ланка (снаббер) з параметрами R_p і C_p .

2.3.4 Моделювання трифазного вимірника струмів і напруг

Виміри в трифазних колах можна виконувати всіма блоками, описаними в п. 2.2.5. Крім того, для трифазного кола існує спеціальний вимірник напруг і струмів усіх трьох фаз – *Three-Phase V-I Measurement* (бібліотека *Measurements*), див. рис. 2.22. Цей блок включається в коло послідовно, у вікні його налаштування (рис. 2.24) вибирається бажана напруга – фазна (*phase-to-ground*) чи лінійна (*phase-to-phase*). Результати виміру можуть бути отримані як у відповідних одиницях системи СІ (вольти, ампер), так і у відносних одиницях (*p.u.*). В останньому випадку потрібно додатково задати базові значення лінійної напруги і потужності.

Виміряні сигнали знімаються з вихідних портів *Vabc* і *Iabc* блока. Існує також можливість «безпроводного» зв'язку вимірника з осцилографом. Для цього у вікні налаштування встановлюють прапорці *Use a label*, а в полях *Signal label*, що з'являються вводять імена змінних для вимірюваних сигналів (наприклад, *V*, *I*). Далі у вікно моделі поміщають два блоки *From* (бібліотека *Simulink/Signal Routing*), у налаштуваннях яких у полі *Goto tag* вказують ті ж імена *V* та *I* відповідно. Виходи блоків *From* підключають

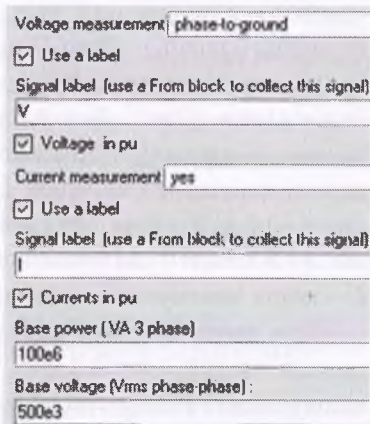


Рисунок 2.24 – Вікно налаштування блока *Three-Phase V-I Measurement*

до осцилографів. Даний спосіб зручний тим, що блоки *From* можна помістити в будь-якому місці вікна моделі. Це дає змогу зберегти її компактність і не захащувати додатковими зв'язками.

Модель трифазного кола з використанням розглянутих вище блоків показана на рис. 2.25.

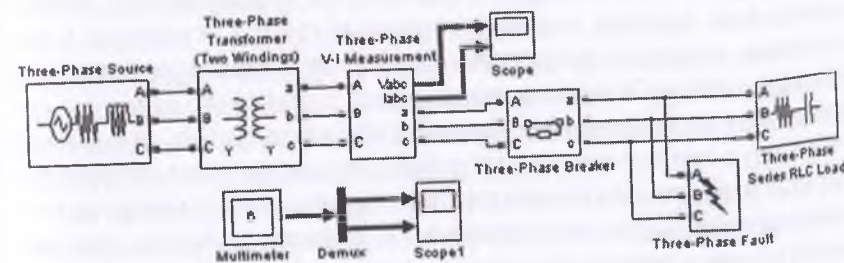


Рисунок 2.25 – Модель трифазного електричного кола

Активно-ємнісне навантаження з'єднане за схемою «трикутник» (у налаштуваннях блока *Three-Phase Series RLC Load* параметр *Configuration* встановлений у значення *Delta*). В момент часу 0,03 с у колі відбувається трифазне коротке замикання (блок *Three-Phase Fault*), а через 0,04 с навантаження відключається автоматичним вимикачем *Three-Phase Breaker*. Графіки перехідних процесів (лінійні напруги і струми), отримані трифазним вимірником, показані на рис. 2.26. Фазні напруги і струми в навантаженні можна виміряти за допомогою блока *Multimeter*, якщо в налаштуваннях блока *Three-Phase Series RLC Load* встановити параметр *Measurements* у значення *Branch voltages and currents*.

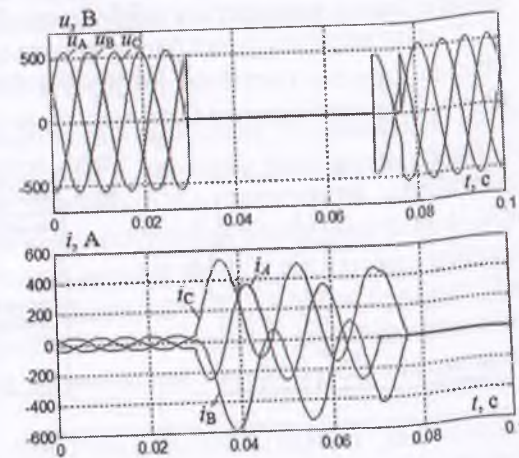


Рисунок 2.26 – Графіки перехідних процесів у трифазному колі

2.4 Моделювання електричних кіл постійного струму в перехідному режимі

Моделювання електричного кола постійного струму в статичному режимі може бути використано як альтернативний метод розрахунку струмів і напруг. Такий спосіб більш наочний, знижує імовірність помилки і скорочує часові витрати в порівнянні з класичними методами розрахунку кіл (законів Кірхгофа, контурних струмів та ін.). Складання такої моделі принципово нічим не відрізняється від моделювання кола синусоїдного струму, за винятком того, що в даному випадку потрібно використовувати джерело постійної напруги. Для візуалізації результатів вимірів можна замість осцилографа *Scope* використовувати індикатор *Display*, оскільки в статичному режимі всі сигнали постійні і не залежать від часу. Однак з погляду моделювання усталений режим у колах постійного струму не становить особливого інтересу. Розглянуті пакети *Simulink* і *Sim Power Systems* призначені для моделювання переважно динамічних процесів. Тому в даному параграфі розглянемо особливості моделювання кіл постійного струму в перехідному режимі, коли стан кола змінюється в результаті комутації.

Розглянемо електричне коло, схема якого наведена на рис. 2.27, а, а модель – на рис. 2.27, б. Модель джерела постійної напруги *DC Voltage Source* знаходиться в бібліотеці *Electrical Sources*.

У вікні налаштування блока задається лише значення напруги. Модель джерела постійного струму в бібліотеці елементів відсутня, але при необхідності його модель можна реалізувати на основі

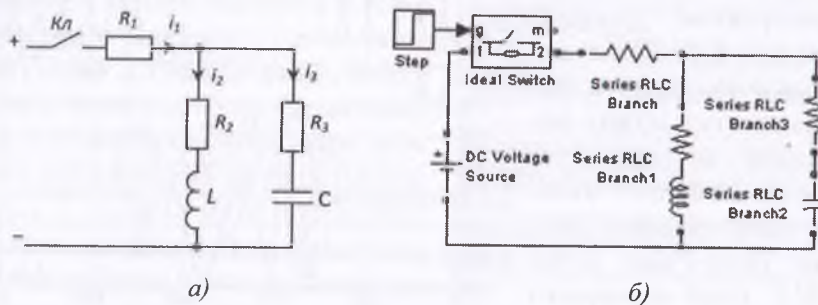


Рисунок 2.27 – Схема (а) і модель (б) електричного кола постійного струму

джерела синусоїдного струму *AC Current Source*, встановивши значення частоти в 0, а фази – у 90° .

Як уже зазначалося в п. 2.2.3, у ролі ключа в колах постійного струму не можна використовувати блок *Breaker* при роботі на розмикання, оскільки для його спрацьовування необхідно, щоб струм через ключ знизився до нуля. У таких випадках застосовується блок *Ideal Switch* (п. 3.1.1). У даному випадку ключ спочатку знаходиться в розімкненому стані, тобто працює на замикання, тому може бути застосований будь-який із зазначених блоків. Керування ключем відбувається за допомогою блока *Step*, сигнал якого в початковий момент часу $t = 0$ приймає значення 1.

Як вихідні сигнали будемо розглядати струм котушки i_2 і напругу конденсатора u . Ці сигнали називають змінними стану електричного кола [6, 14], оскільки вони обумовлюють запас енергії в колі й однозначно характеризують його стан у кожен момент часу. Для виміру використовується блок *Multimeter* (на рис. 2.27, б не показаний). У даному випадку нас цікавить напруга не на всій *RC*-ланці, а лише на конденсаторі, тому для нього потрібно використовувати окремий блок *Series RLC Branch*, у вікні налаштування якого параметр *Measurement* треба встановити в значення *Branch voltage*. Резистор R_3 моделюється також окремим блоком. Для елементів L і R_2 можна використовувати один блок, параметр *Measurement* якого встановити в значення *Branch current*.

2.4.1 Установка початкових умов

Якщо в моделі використовується ключ, то задавати початкові умови (п. у.) немає необхідності, оскільки вони розраховуються програмою автоматично в залежності від стану ключа до комутації (параметр *Initial state* у вікні налаштування блока *Ideal Switch*). Але якщо комутація відбувається в момент часу $t = 0$, то можна обійтися без ключа, щоб не захаращувати модель додатковими блоками. При цьому готова модель сприймається програмою як електричне коло після комутації, а її стан до комутації визначається початковими умовами.

Для задання п. у. в *Sim Power Systems* існують два способи. Найпростіший – це ввести відповідні початкові значення для всіх змінних стану кола (струмів котушок і напруг конденсаторів) безпо-

середньо у вікні налаштування блоків *Series (Parallel) RLC Branch*. Для цього попередньо треба установити прапорець *Set the initial inductor current* (для котушки) чи *Set the initial capacitor voltage* (для конденсатора). Але ці параметри присутні лише в блоках останніх версій MATLAB. Другий спосіб – задавання початкових умов за допомогою блока *Powergui* (кнопка *Initial States Setting*). Після запуску цього інтерфейсу відкривається вікно, у правій частині якого пропонується задати п. у. змінних стану рівними сталим значенням (*To Steady States*), нулю (*To Zero*) або ввести конкретні значення в поле *Set Selected States* (попередньо вибравши необхідний струм чи напругу зі списку в центрі вікна). В останній версії програми додатково пропонується пункт *To Block Settings*, тобто задати п. у. відповідно до настроювань блоків, що моделюють котушки і конденсатори. Вибір підтверджується натисканням кнопки *Close*.

Якщо п. у. не задані, то вони автоматично приймають значення, які відповідають сталому стану кола. У цьому випадку в колі без ключа спостерігати перехідний процес не вдасться, оскільки всі сигнали будуть мати постійне значення протягом усього часу моделювання.

Перехідний процес у досліджуваному колі з нульовими п. у. показаний на рис. 2.28, а (ключ працює на замикання).

Розглянемо послідовність дій для моделі без блока *Ideal Switch*, якщо потрібно використовувати розмикаючий ключ. У цьому випадку усталені значення сигналів у колі при замкнутому ключі є

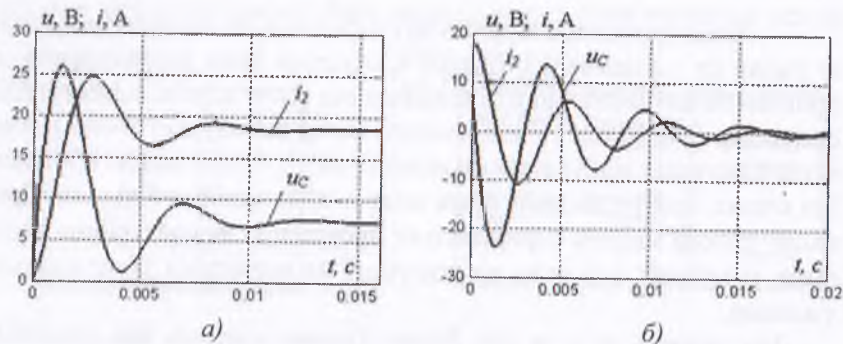


Рисунок 2.28 – Графіки перехідних процесів у колі постійного струму: а) при замиканні ключа; б) при розмиканні ключа

початковими умовами для того ж кола, у якому ключ працює на розмикання. Після того, як модель зібрана повністю (з перемичкою замість ключа), потрібно:

1) провести сеанс моделювання з нульовими п. у. і зняти з осцилограм сталі значення струму котушки $I_{L_{ст}}$ і напруги конденсатора $U_{C_{ст}}$; у даному випадку це значення 18,5 А та 7,4 В відповідно (як видно з рис. 2.28, а);

2) встановити прапорці в блоках котушки та конденсатора і ввести значення $I_{L_{ст}}$ та $U_{C_{ст}}$ в поля для початкових умов;

3) розімкнути гілку, в якій має бути ключ (видалити лінію між джерелом і резистором R_1);

4) запустити виконання моделі.

Перехідний процес з ненульовими п. у. показаний на рис. 2.28, б. Описаний вище алгоритм застосовний для одноразової комутації. Якщо ж потрібно здійснити кілька замикань і розмикань ключа протягом одного сеансу моделювання, то треба застосувати блок *Ideal Switch*, керований блоком *Timer*.

2.4.2 Структурна модель кола

Модель на рис. 2.27, б є так званою віртуальною моделлю електричного кола, оскільки при її складанні немає необхідності звертатися до математичного опису кола і розв'язувати диференціальні рівняння. При цьому всі блоки і з'єднання в моделі імітують елементи реального електричного кола (джерело, резистори, проводи та ін.). Однак існує інший підхід до моделювання – складання структурної моделі, усі компоненти якої відображають математичні зв'язки в системі диференціальних рівнянь, що описують коло. У такій моделі використовуються тільки блоки пакета *Simulink*, переважно з бібліотек *Math Operations* і *Continuous*.

Для досліджуваного кола (рис. 2.27, а) запишемо систему рівнянь на підставі законів Кірхгофа для стану після комутації, тобто при замкнутому ключі:

$$u_L + i_1 R_1 + i_2 R_2 = E;$$

$$u_C + i_1 R_1 + i_3 R_3 = E;$$

$$i_1 = i_2 + i_3,$$

$$\text{де } u_L = L \frac{di_2}{dt}, \quad i_3 = C \frac{du_C}{dt}.$$

Перепишемо дану систему у вигляді, більш зручному для моделювання:

$$L \frac{di_2}{dt} = E - i_1 R_1 - i_2 R_2;$$

$$i_3 R_3 = E - i_1 R_1 - u_C;$$

$$i_1 = i_2 + i_3.$$

Цій системі рівнянь відповідає структурна *Simulink*-модель на рис. 2.29. Коефіцієнти представлені підсилювальними блоками *Gain*, дві інтегруючі ланки свідчать про те, що досліджуване коло є системою другого порядку. Усі сигнали на входах верхнього і нижнього суматорів моделюють доданки першого і другого рівнянь відповідно, а середній суматор моделює третє рівняння – суму струмів у вузлі кола. Значення всіх змінних у блоках моделі задаються за допомогою *m*-файлу (докладніше див. п. 1.1.2) Графіки перехідних процесів ідентичні отриманим за допомогою віртуальної моделі при нульових початкових умовах (рис. 2.28, а).

Якщо потрібно моделювати перехідний процес при розмиканні ключа, то треба змінити систему рівнянь відповідно до стану кола після комутації (у даному випадку гілка з джерелом виявиться розімкнутою, і коло перетвориться в одноконтурне). При цьому зміниться і сама структурна модель. Крім того, треба задати ненульові початкові умови у вікнах налаштування блоків *Integrator* (параметр *Initial condition*). Результати моделювання ідентичні зображеним на рис. 2.28, б.

Таким чином, обидві моделі – віртуальна і структурна – є розв'язком однієї і

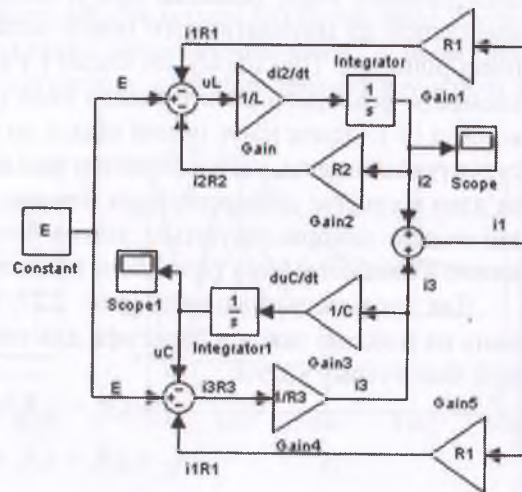


Рисунок 2.29 – Структурна модель електричного кола постійного струму

тєї ж системи диференціальних рівнянь, що описують електричне коло. Різниця між ними полягає в тому, що у віртуальній моделі (рис. 2.27, а) система рівнянь присутня в неявному вигляді і реалізується програмою автоматично при з'єднанні віртуальних елементів кола – блоків пакета *Sim Power Systems*, тоді як у структурній моделі (рис. 2.27, б) дослідник самостійно складає математичний опис, а кожен блок пакета *Simulink* реалізує певну математичну операцію у відповідності з цим описом.

Варто підкреслити, що структурне моделювання можна застосовувати і для кіл змінного струму. У цьому випадку також складається система диференціальних рівнянь для миттєвих значень струмів і напруг, а як джерело сигналу використовується блок *Sine Wave* або *Signal Generator*.

2.5 Моделювання несинусоїдних електричних кіл

Розглянемо особливості моделювання і дослідження кіл несинусоїдного струму на прикладі простого електричного кола, що складається з послідовно з'єднаних резистора, котушки і конденсатора, на вхід якого подається напруга прямокутної форми. Його модель наведена на рис. 2.30, а.

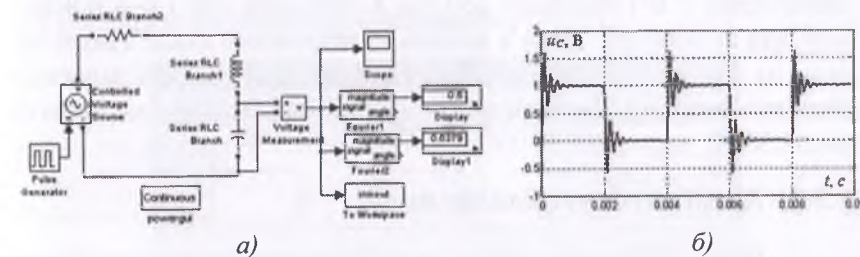


Рисунок 2.30 – Модель несинусоїдного електричного кола (а) й осцилограма напруги на конденсаторі (б)

2.5.1 Моделювання джерел несинусоїдних сигналів

У пакеті *SPS* немає готової моделі джерела несинусоїдної напруги чи струму, тому при необхідності таке джерело потрібно скласти самостійно, використовуючи блоки *Simulink*. А оскільки для моделювання віртуального електричного кола застосовується пакет *SPS*, то елементи двох пакетів необхідно з'єднати в одній

моделі за допомогою спеціальних перехідних блоків – керованих джерел напруги і струму: *Controlled Voltage Source* і *Controlled Current Source*, бібліотека *Electrical Sources* пакета *SPS* (рис. 2.31). Вони мають два типи портів – для *Simulink*-сигналу (вхід *s*) і *SPS*-сигналу (порти «+» і «-»). Таким чином, ці блоки служать для перетворення безрозмірного сигналу в еквівалентну напругу чи струм, тобто являють собою інтерфейс між двома пакетами. У ролі такого інтерфейсу блок *Controlled Voltage Source* був використаний у моделі трифазного синусоїдного джерела зі змінюваними параметрами (п. 2.3.1.3, рис. 2.21, а). У даному випадку разом із блоком *Pulse Generator* він являє собою модель джерела напруги прямокутної форми (рис. 2.30, а).

На вхід *s* блока *Controlled Voltage Source* подається *Simulink*-сигнал від генератора прямокутних імпульсів *Pulse Generator* з амплітудою 1 В і частотою 250 Гц, а затискачі «+» і «-» призначені для включення блока в модель електричного кола. Таким чином, на виході утворюється напруга такої форми, яка задається *Simulink*-сигналом. Осцилограма напруги конденсатора показана на рис. 2.30, б.

2.5.2 Аналіз несинусоїдного кола

Пакет *SPS* пропонує різні можливості аналізу несинусоїдних електричних кіл. Для виміру амплітуди і фази заданої гармоніки вихідного сигналу використовується блок *Fourier*, що був розглянутий у п. 2.2.5.4. Так, у досліджуваній моделі на рис. 2.30, а постійна складова напруги конденсатора вимірюється блоком *Fourier1* (вона дорівнює 0,5 В, номер гармоніки *n* встановлений у значення 0), а амплітуда основної гармоніки – блоком *Fourier2* (0,638 В, *n* = 1). У полі *Fundamental frequency* обох блоків встановлене значення частоти основної гармоніки – 250 Гц. Результати вимірів відображаються блоками *Display*.

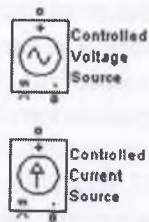


Рисунок 2.31 – Блоки *Controlled Voltage Source* і *Controlled Current Source*

Частотний спектр сигналу можна одержати за допомогою блока *Powergui* (утиліта *FFT Analysis*). Насамперед, необхідно подати досліджуваний сигнал на вхід блока *To Workspace*, параметр *Save format* якого встановити в значення *Structure With Time*, і виконати сеанс моделювання. Далі – запустити інтерфейс *FFT Analysis* і у вікні, що відкрилося (рис. 2.32), праворуч встановити значення частоти основної гармоніки в герцах (поле *Fundamental Frequency*).

Інші параметри можна залишити заданими за замовчуванням. Після натискання кнопки *Display* з'явиться графік частотного спектра, що являє собою сукупність гармонічних складових сигналу. Висота кожної лінії на графіку пропорційна амплітуді, а місце на осі абсцис визначається частотою даної гармоніки. У налаштуваннях вікна можна вибрати стиль відображення діаграми: по осі абсцис – частота чи номер гармоніки (поле *Frequency axis*), по осі ординат – відсотки чи відносні одиниці (поле *Display style*), а також ширину спектра (поле *Max Frequency*). Як видно з рис. 2.32, у спектрі напруги на конденсаторі присутні тільки непарні гармоніки і постійна складова (значення якої 0,5 В відповідає вимірюваному блоком *Fourier1* на рис. 2.30, а). Частотний спектр можна представити не тільки графічно, але й у числовому вигляді (параметр *Display style* встановити в значення *List...*). У вікні також відображається значення коефіцієнта гармонічних спотворень (*THD*).

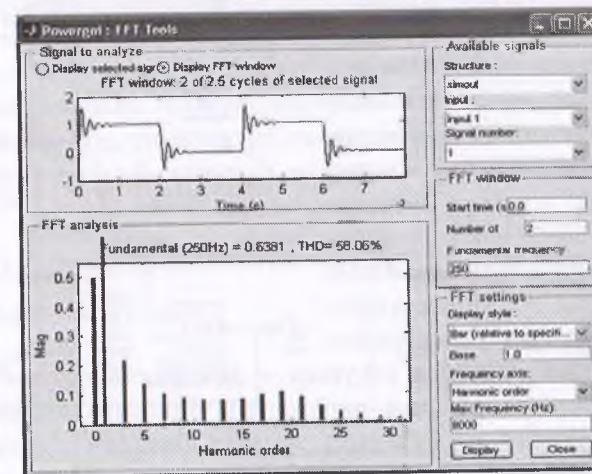


Рисунок 2.32 – Вікно інтерфейсу *FFT Analysis*

За допомогою розглянутого раніше інтерфейсу *LTI Viewer* блока *Powergui* (п. 2.2.6.2) можна досліджувати поведінку даного електричного кола при різних частотах (частотні характеристики) і виявляти резонансні режими як на основній, так і на вищих гармоніках вхідного несинусоїдного сигналу.

2.6 Моделювання нелінійних електричних кіл

Усі блоки пакета *SPS*, призначені для моделювання резистивно-індуктивно-ємнісних кіл (п. 2.2.2), являють собою моделі лінійних елементів, тобто їхні параметри не залежать від значень струмів і напруг, а вольтамперні характеристики (ВАХ) є лінійними. При необхідності моделювати коло з нелінійним елементом його модель треба скласти самостійно. При цьому характеристика такого елемента може бути задана одним із двох способів – у табличній або в аналітичній формі.

2.6.1 Модель елемента, заданого в табличній формі

Розглянемо табличний спосіб задання характеристики нелінійних елементів на прикладі кола синусоїдного струму з нелінійною котушкою індуктивності (рис. 2.33, а). Котушка з номінальною індуктивністю $L = 2$ Гн призначена для роботи при номінальній напрузі 120 В і номінальній частоті мережі 60 Гц. При підвищенні напруги індуктивність зменшується до 0,4 Гн, відповідно зменшується й індуктивний опір котушки змінного струму.

Для спрощення вважаємо індуктивність постійною в межах кожної ділянки. Отже, вебер-амперна характеристика котушки – залежність потокозчеплення ψ^* від струму I^* в обмотці, виражена у відносних одиницях – являє собою ламану, яка складається з двох ділянок з різним нахилом (рис. 2.34). При цьому за базові значення струму I_{base} і потокозчеплення ψ_{base} використані їхні номінальні значення:

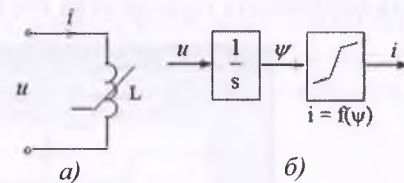


Рисунок 2.33 – Коло з нелінійною котушкою:

а) принципова схема; б) структурна схема

$$I^* = \frac{I}{I_{base}}, \quad \psi^* = \frac{\psi}{\psi_{base}},$$

$$\text{де } I_{base} = \frac{U_H}{X_H} = \frac{U_H}{2\pi f_H L} = 0,159 \text{ А}, \quad \psi_{base} = I_H L_H = 0,159 \cdot 2 = 0,318 \text{ Вб}.$$

Індуктивність кожної ділянки характеристики визначається співвідношенням

$$L = \frac{\Delta \psi^* \cdot \psi_{base}}{\Delta I^* \cdot I_{base}},$$

де $\Delta \psi^*$ і ΔI^* – приріст потокозчеплення і струму (див. рис. 2.34).

Зв'язок струму, потокозчеплення і напруги на затискачах котушки описується співвідношеннями

$$u = L \frac{di}{dt} = \frac{d\psi}{dt}, \quad \psi = \int u dt \quad i = \frac{\psi}{L},$$

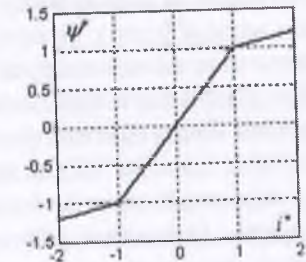


Рисунок 2.34 – Характеристика $\psi^* = f(i^*)$ нелінійної котушки індуктивності

а також структурною схемою на рис. 2.33, б.

Таким чином, нелінійна індуктивність може бути представлена у вигляді керованого джерела струму, струм якого є нелінійною функцією потокозчеплення і, отже, напруги на котушці.

Модель котушки, виконана у вигляді підсистеми *Nonlinear Inductance*, зображена на рис. 2.35, а. Напруга між вхідним і вихідним затискачами вимірюється блоком *Voltage Measurement* і надходить на вхід інтегруючої ланки, яка перетворює цю напругу в магнітний потік. Нелінійний зв'язок $i = f(\psi)$ реалізований за до-

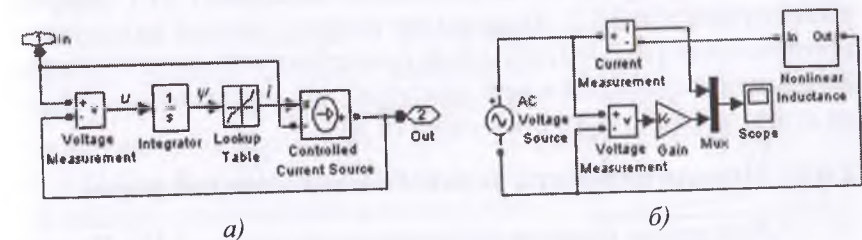


Рисунок 2.35 – Модель нелінійної котушки індуктивності *Nonlinear Inductance* (а); модель електричного кола з нелінійною котушкою (б)

помогою блока *Lookup Table* (бібліотека *Simulink/Lookup Tables*), що служить для задавання функції однієї змінної в табличній формі. Цей блок забезпечує лінійну інтерполяцію й екстраполяцію для входніх сигналів, рівні яких мають проміжні значення (знаходяться між вузловими точками таблиці) або навіть виходять за межі заданого інтервалу зміни аргументу. Вікно параметрів блока містить поля для введення значень входнього (*Vector of input values*) і вихідного (*Table data*) векторів даних. Вихідний вектор може містити як числові дані, так і записи різних функцій MATLAB. У даному випадку входній вектор заданий значеннями поточкозчеплення ψ : $[-1,2 \ -1 \ 1 \ 1,2] \cdot 0,318$, а вихідний – відповідними значеннями струму котушки I , що визначаються характеристикою з рис. 2.34: $[-2 \ -1 \ 1 \ 2,5] \cdot 0,159$.

Далі сигнал надходить на керуючий *Simulink*-вхід джерела *Controlled Current Source*, що створює струм у зовнішньому колі відповідно до цього сигналу. Модель котушки *Nonlinear Inductance* включена в зовнішнє коло (рис. 2.35, б) за допомогою портів *In* та *Out*.

Напруга синусоїдного джерела складає 180 В, тобто вище номінальної. Осцилограми струму і напруги в колі наведені на рис. 2.36. Видно, що форма струму відрізняється від синусоїдної, оскільки його значення потрапляють в область насичення характеристики. У цьому випадку нелінійний елемент є генератором вищих гармонік. Досліджувати спектральний склад струму можна за допомогою інтерфейсу *FFT Analysis*, розглянутого в п. 2.5.2. Якщо входню напругу знизити до номінального значення 120 В, то струм буде синусоїдним. Крім того, ці осцилограми показують, що в колі присутня саме індуктивність, оскільки струм за фазою відстає від напруги на 90° .

2.6.2 Модель елемента, заданого в аналітичній формі

Розглянемо схему стабілізатора напруги (рис. 2.37). Призначення даного пристрою – підтримувати постійною або майже постійною вихідну напругу U_H при впливі різних дестабілізуючих

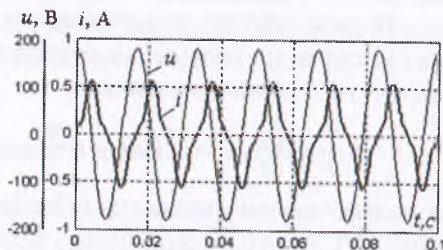


Рисунок 2.36 – Осцилограми в колі з нелінійною котушкою

факторів (коливанні напруги живлення, зміні опору навантаження R_H та ін.). Для забезпечення незмінності напруги в схему паралельно навантаженню включено нелінійний опір (НО), вольтамперна характеристика якого наведена на рис. 2.38. Аналітичні вирази, що описують кожну з ділянок ВАХ, мають наступний вигляд:

$$I = 0,02 U \text{ при } 0 < U < 5 \text{ В;}$$

$$I = 0,5 U - 2,4 \text{ при } U \geq 5 \text{ В.}$$

Модель НО, виконана у вигляді підсистеми *Nonlinear resistance*, показана на рис. 2.39, а. Кожен алгебраїчний вираз реалізується блоком *Fcn* (бібліотека *Simulink/User-Defined Functions*). Цей блок розраховує значення функції відповідно до формули, введенної в поле *Expression* вікна налаштування. Як аргумент використовується входній сигнал, що позначається змінною u . Переключення між ділянками ВАХ (між аналітичними виразами, що описують ці ділянки) при напрузі 5 В здійснюється за допомогою блока *Switch*, розглянутого раніше в п. 1.2.2.1. Параметр *Threshold* блока *Switch* встановлений у значення 5, а на керуючий вхід подається напруга U_H на затискачах НО. Якщо $U_H < 5$ В, то на вихід блока надходить сигнал з нижнього входу, тобто реалізується полого ділянка ВАХ, інакше сигнал надходить з верхнього входу (прямовисна ділянка ВАХ). Відповідно до розрахованого сигналу струм у колі створюється керованим джерелом струму *Controlled Current Source*, аналогічно п. 2.6.1.

Модель стабілізатора з нелінійним опором *Nonlinear resistance* показана на рис. 2.39, б. У моделі встановлені наступні значення параметрів: $U_{вх} = 15$ В, $R_6 = 25$ Ом, $R_H = 50$ Ом. Оскільки в даному випадку розглядається коло постійного струму, то замість осцилографів *Scope* для візуалізації результатів моделювання використовується індикатор *Display*.

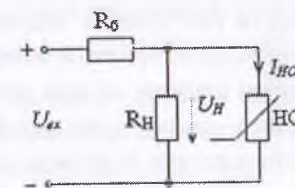


Рисунок 2.37 – Схема стабілізатора напруги

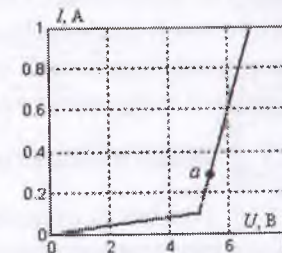


Рисунок 2.38 – ВАХ нелінійного опору

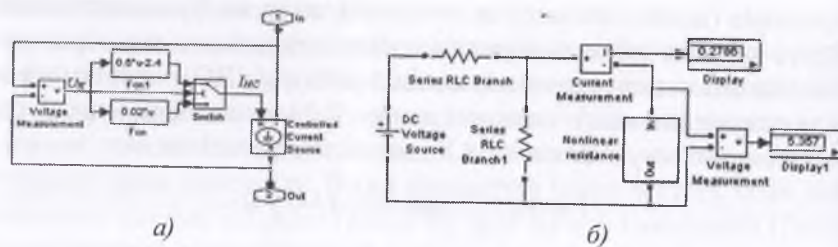


Рисунок 2.39 – Модель нелінійного опору *Nonlinear resistance* (а); модель стабілізатора напруги з нелінійним опором (б)

Значення напруги і струму нелінійного елементу ($U_H = 5,35$ В, $I_{HO} = 0,28$ А) при даних параметрах моделі відповідають робочій точці а, позначеній на рис. 2.38. При зміні вхідної напруги $U_{BX} = 10 \dots 25$ В або опору навантаження $R_H \geq 18$ Ом напруга U_H змінюється дуже незначно, залишаючись у межах 5...6 В. При $U_{BX} < 10$ В чи $R_H < 18$ Ом напруга навантаження різко падає, що відповідає пологій ділянці ВАХ. Тобто при даних значеннях U_{BX} і R_H стабілізатор не виконує свою функцію підтримки постійності вихідної напруги.

Для побудови ВАХ нелінійного опору (рис. 2.38), потрібно провести ряд сеансів моделювання при різних U_{BX} або R_H і одержати відповідні значення напруги U_H і струму I_{HO} через НО. За отриманими даними можна побудувати залежність $I_{HO} = f(U_H)$. Інший варіант – скласти окрему *Simulink-модель*, у якій джерело напруги U_H у підсистемі *Nonlinear resistance* (рис. 2.39, а) замість блока *Voltage measurement* використати джерело лінійно наростаючого впливу *Ramp*. При цьому сигнал на виході блока *Switch* буде відповідати струму через НО I_{HO} (блок *Controlled Current Source* у даному випадку не потрібний). Обидва сигнали – струм і напругу – подати на віртуальний графопобудовувач *XY Graph*, налаштувавши відповідним чином параметри осей. При запуску моделі в окремому вікні з'явиться побудована залежність $I_{HO} = f(U_H)$.

2.7 Моделювання нестаціонарних електричних кіл

Більшість стандартних блоків, застосовуваних для моделювання електричних кіл, є моделями стаціонарних елементів, тобто їхні параметри не змінюються за часом. Винятком є блоки *Breaker*, *Ideal Switch*, а також моделі силових напівпровідникових приладів,

що працюють у режимі керованого ключа (*IGBT*, *MOSFET*, див. п. 3.1.1). Їхній опір у залежності від керуючого сигналу може приймати значення 0 чи ∞ .

Якщо виникає необхідність моделювання нестаціонарного кола з резистивним, індуктивним або ємнісним елементом, параметр якого змінюється за часом, то модель такого елемента потрібно скласти самостійно. На рис. 2.40, а наведена модель електричного кола синусоїдного струму з нестаціонарним елементом, виконаним у вигляді підсистеми *Var_param*. Як приклад розглянемо два варіанти реалізації цієї підсистеми:

- 1) нестаціонарний резистор (рис. 2.40, б);
- 2) нестаціонарний конденсатор (рис. 2.40, в).

У момент часу 0.05 с опір резистора змінює своє значення з 2 до 4 Ом, а ємність конденсатора – з 1300 до 663 мкФ (що відповідає зміні ємнісного опору також з 2 до 4 Ом при частоті джерела 60 Гц). Обидві моделі побудовані на основі вимірника напруги *Voltage Measurement*, який перетворює *SPS*-сигнал у *Simulink*-сигнал.

Далі в моделі резистора цей сигнал u ділиться блоком *Divide* на значення опору R , а в моделі конденсатора відбувається його диференціювання і множення блоком *Product* на значення ємності C . Тим самим реалізуються залежності: $i = \frac{u}{R}$ – для резистора та $i = C \frac{du}{dt}$ – для конденсатора. Отримане в результаті значення струму i надходить на кероване джерело струму, яке двома затискачами *Port1* і *Port2* включається в зовнішнє коло. При цьому відбувається обернене перетворення *Simulink*-сигналу в *SPS*-сигнал. Змінювання значень R і C у заданий момент часу здійснюється за допомогою блока

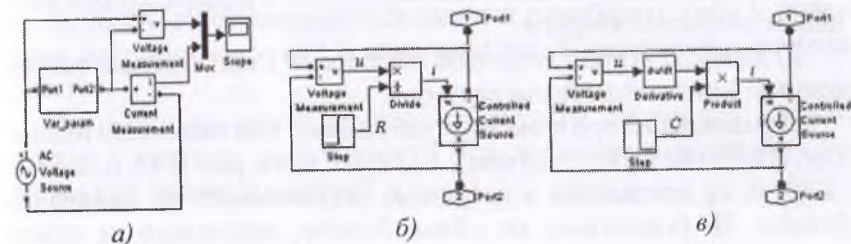


Рисунок 2.40 – Модель кола з нестаціонарним елементом *Var_param* (а); модель нестаціонарного резистора (б); модель нестаціонарного конденсатора (в)

Step. Якщо потрібно реалізувати змінювання $R(t)$ і $C(t)$ за лінійною залежністю, то треба використовувати блок *Ramp*.

Осцилограми для двох варіантів моделі наведені на рис. 2.41: i_R – струм у колі з нестационарним резистором (див. рис. 2.40, б); i_C – струм у колі з нестационарним конденсатором (див. рис. 2.40, в); u – напруга на елементі *Var_param*. Видно, що при збільшенні опору (як резистивного, так і ємнісного) струм зменшується, причому в колі з резистором він збігається по фазі з напругою, а в колі з конденсатором – випереджає напругу на 90° .

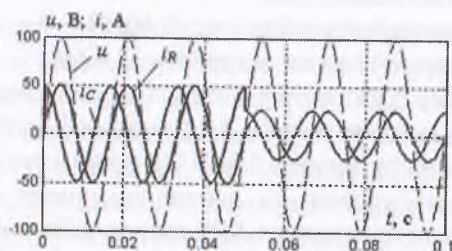


Рисунок 2.41 – Графіки переходних процесів у нестационарному колі

2.8 Сумісність елементів електричних кіл

При моделюванні *RLC*-кіл необхідно пам'ятати про основні принципи з'єднання ідеальних конденсаторів, котушок і джерел, оскільки деякі сполучення елементів у *MATLAB* є неприпустимими і призводять до видачі повідомлення про помилку. До неприпустимих з'єднань відносяться наступні:

- 1) джерело напруги включене паралельно з конденсатором або з іншим джерелом напруги;
- 2) джерело струму включене послідовно з індуктивною котушкою або з іншим джерелом струму.

Повідомлення про помилку буде видано і при включенні підсистем, аналогічних розглянутим у п. 2.6, 2.7 (див. рис. 2.35, а, 2.39, а, 2.40, б, в), послідовно з котушкою індуктивності чи вимикачем *Breaker*. Ці підсистеми, як і блок *Breaker*, реалізовані на основі джерела струму, тому на них також поширюються зазначені вище обмеження. Для запобігання помилки потрібно зашунтувати джерело струму чи індуктивну котушку резистором з великим опором

(порядку 1 МОм), тобто замінити послідовне з'єднання змішаним. Опір має бути великим, щоб паралельна ділянка не створювала істотного впливу на іншу частину кола. Якщо в колі послідовно включено блок *Breaker*, то додаткове шунтування не потрібне, оскільки в моделі даного блока вже передбачена паралельна *RC*-ланка – снаббер (п. 2.2.3).

Аналогічно, при паралельному з'єднанні джерела напруги з іншим джерелом напруги чи конденсатором потрібно замінити паралельне з'єднання змішаним, тобто включити послідовно з одним із елементів резистор з малим опором.

Контрольні питання до розділу 2

1. Чим відрізняються функціональні моделі пакета *Simulink* від віртуальних моделей пакета *Sim Power Systems*?
2. Як установити параметри блока *Series RLC Branch* для одержання моделі *RL*-кола? *RC*-кола? *LC*-кола?
3. Чому блок *Breaker* не можна використовувати при роботі на розмикання в моделях кіл постійного струму?
4. З якою метою в складі блока *Breaker* застосовується *RC*-ланка (снаббер)?
5. Як виміряти діюче значення струму чи напруги в моделі електричного кола?
6. Як у *MATLAB* побудувати амплітудно-частотну і фазочастотну характеристики електричного кола?
7. Як одержати передатну функцію електричного кола, модель якого складена з елементів пакета *Sim Power Systems*?
8. Як реалізувати в моделі режим короткого замикання?
9. Яким чином виконується задання початкових умов у моделі кола постійного струму? В яких випадках у цьому немає необхідності?
10. Чому не можна в моделі з'єднувати послідовно джерело струму і котушку індуктивності? Які додаткові підключення потрібні при такому з'єднанні?

Завдання для самостійного виконання

1. Скласти модель підсистеми, яка виконує вимірювання активної потужності трифазного кола методом двох ватметрів [11]. Використати блок *Active & Reactive Power*.

2. Скласти модель нестационарної котушки, індуктивність якої плавно зростає з 2 до 6 мГн протягом 0,1 с. Реалізувати модель у вигляді підсистеми *Var_param* і включити її в коло аналогічно рис. 2.40, а.

3. За допомогою інтерфейсу *LTI Viewer* одержати передатну функцію кола, схема і модель якого наведені на рис. 2.1.

4. Скласти модель трифазного кола, в якому у визначений момент часу відбувається переключення споживача зі схеми «трикутник» на схему «зірка».

5. Скласти математичний опис і структурну модель кола на рис. 2.27, а при роботі ключа на розмикання (див. п. 2.4.2).

Розділ 3

МОДЕЛЮВАННЯ НАПІВПРОВІДНИКОВИХ ПЕРЕТВОРЮВАЧІВ

Пакет *Sim Power Systems*, використовуваний разом з *Simulink*, надає широкі можливості з моделювання силових напівпровідникових перетворювачів і систем керування. Віртуальні моделі, розроблені на основі цих двох пакетів, дозволяють більш глибоко вивчити фізичні процеси перетворення електроенергії, не вдаючись до використання реальних напівпровідникових пристроїв. У даному розділі розглядаються особливості стандартної елементної бази пакета *Sim Power Systems*, призначеної для моделювання силових перетворювачів, а також основні принципи моделювання систем керування.

3.1 Елементи силових електроніки

3.1.1 Стандартні моделі елементів бібліотеки Power Electronics

Базові елементи, призначені для моделювання напівпровідникових пристроїв, знаходяться в бібліотеці Power Electronics. Їхні найменування, зовнішній вигляд, вольтамперні характеристики й алгоритм функціонування наведені в табл. 3.1. Тут *On* означає включений, або провідний, стан приладу, *Off* – виключений, або замкнений.

Розглянемо основні особливості блоків табл. 3.1. Більшість з них складаються на основі спрощеної моделі, зображеної на рис. 3.1. До її складу входить ключ *SW*, керований блоком логіки, внутрішній опір R_{on} та індуктивність L_{on} напівпровідникового пристрою у відкритому стані, а також джерело постійної напруги V_f , що моделює нелінійність вольтамперної характеристики. У моделі кожного блока передбачене підключення паралельної *RC*-ланки (навантаження) з параметрами R_s і C_s , призначені для гасіння перенапруг, що

Таблиця 3.1 – Моделі напівпровідникових елементів у бібліотеці Power Electronics

Піктограма блоку	Назва	Позначення на схемі	Характеристика	Умови для переходу в стан	
				On	Off
1	2	3	4	5	6
	Діод			$V_{ak} > V_f$	$V_{ak} \leq V_f$ або $I_{ak} = 0$
	Тиристор (спрощена і повна моделі)			$g > 0$ і $V_{ak} > V_f$ ($I_{ak} > I_l$)	$V_{ak} \leq V_f$ або $I_{ak} = 0$ або $I_{ak} < I_l$, $g = 0$
	Керований тиристор			$g > 0$ і $V_{ak} > V_f$	$g = 0$ або $V_{ak} \leq V_f$
	Транзистор IGBT			$g > 0$ $V_{CE} > V_f$	$g = 0$ або $V_{CE} \leq V_f$
	Транзистор MOSFET			$V_{DS} \geq 0$ $g > 0$ або $V_{DS} < 0$ $ V_{DS} > V_f$ (ч/з діод)	$V_{DS} \geq 0$ $g = 0$ або $V_{DS} < 0$ $I_d = 0$
	Ідеальний ключ			$g > 0$	$g = 0$

виникають у процесі комутації. Нахил вольтамперної характеристики визначається опором R_{on} , початкове значення струму через прилад – параметром I_C вікна налаштування. На рис. 3.2 як приклад показане вікно для блоків *Diode* і *Thyristor*. Інші блоки табл. 3.1 містять додаткові параметри, про які буде сказано далі.

Всі елементи, крім діода, мають керуючий вхід g , на який подається *Simulink*-сигнал 0 або 1. У залежності від цього сигналу, а також від струму і напруги між вхідним і вихідним затискачами, елемент переходить у відкритий *On* або закритий *Off* стан, що моделюється відповідно замкнутим або розімкнутим ключем *SW*. Умови, необхідні для переходу блоків з одного стану в інший, наведені в табл. 3.1. Так, наприклад, для включення тиристора необхідно подати сигнал 1 на вхід g , при цьому напруга між анодом і катодом V_{ak} повинна бути більшою, ніж гранична напруга V_f . Запирання тиристора відбудеться при зниженні струму через нього до 0 або при зменшенні напруги до значення $V_{ak} \leq V_f$.

Для уточненої моделі тиристора *Detailed Thyristor* існують два додаткових параметри – струм включення I_l (*Latching current*) і час відновлення запираючих властивостей T_q (*Turn-off time*). У версії MATLAB 2008b параметри I_l і T_q спочатку не відображаються у вікні налаштування, тому що за замовчуванням $L_{on} = 0$. Для їхньої активізації потрібно задати значення L_{on} ,

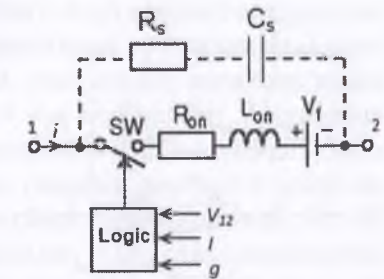


Рисунок 3.1 – Внутрішня модель блоків бібліотеки *Power Electronics*

Resistance R_{on} (Ohms):
0.001

Inductance L_{on} (H):
0

Forward voltage V_f (V):
0.8

Initial current I_c (A):
0

Snubber resistance R_s (Ohms):
500

Snubber capacitance C_s (F):
250e-9

☐ Show measurement port

Рисунок 3.2 – Вікно налаштування блоків *Diode* і *Thyristor*

що відрізняється від нуля, і натиснути кнопку *Apply*. Для надійного вмикання тиристора необхідно, щоб сигнал керування $g > 0$ подавався протягом усього часу, поки струм через прилад не досягне значення I , інакше при $g = 0$ тиристор знову перейде в закритий стан. Перебуваючи у відкритому стані, він закриється автоматично при $g = 0$ у тому випадку, якщо струм зменшиться до значення $I_{ak} < I$. Для запирання тиристора треба подавати від'ємну напругу між анодом і катодом V_{ak} як мінімум протягом часу T_q , інакше, якщо V_{ak} знову стане додатним за час, менший від T_q , тиристор автоматично відкриється, навіть якщо $g = 0$.

Наступні дві моделі – керований тиристор *GTO* і транзистор *IGBT* – можуть бути як відкриті, так і закриті в будь-який момент часу за допомогою логічного сигналу на вході g , тобто є повністю керованими пристроями. Вони мають два додаткових параметри – T_f (*fall time*) і T_t (*tail time*). При $g = 0$ прилад починає закриватися (якщо до цього він перебував у провідному стані), однак струм не припиняється миттєво. Протягом часу T_f він зменшується до 10% від свого значення I_n у момент відключення, а потім за час T_t знижується до нуля (рис. 3.3). Блок *IGBT/Diode* являє собою спрощений варіант блока *IGBT*: має тільки три параметри налаштування – R_{on} , R_s , C_s , але містить у своєму складі зворотний паралельний діод. Цей блок призначений для моделювання схем перетворювачів напруги, у яких силові модулі *IGBT* шунтуються діодами з метою забезпечити шлях проходження струму навантаження при закритих ключах.

Модель транзистора *MOSFET* також містить зворотний паралельний діод, наявність якого обумовлена технологічними особливостями виготовлення даних приладів. При цьому сам транзистор характеризується тільки опором R_{on} , а параметри L_{on} , R_d , V_f – це відповідно індуктивність, опір і гранична напруга паразитного діода. Внутрішня модель *MOSFET*-блока зображена на рис. 3.4. Як видно з табл. 3.1, цей транзистор може бути відкритий як при до-

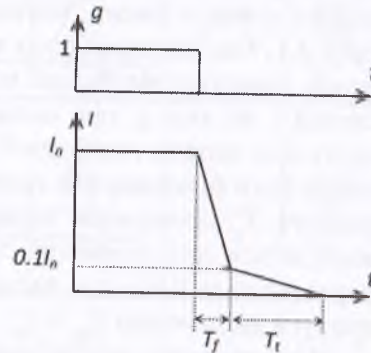


Рисунок 3.3 – Графік зміни струму в моделях *GTO* и *IGBT*

додатній, так і при від'ємній напрузі між стоком і джерелом V_{DS} . При $V_{DS} \geq 0$ він керується так само, як *IGBT* і *GTO* – сигналом на вході g . Якщо $V_{DS} < 0$, то струм проходить через зворотний діод незалежно від керуючого сигналу, і в цьому випадку алгоритм роботи *MOSFET* визначається алгоритмом діода, який відкритий при напрузі більшій, ніж гранична V_r , а закривається при зниженні струму до нуля.

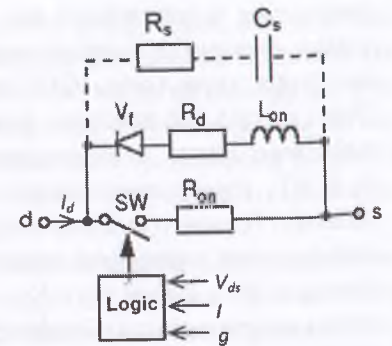


Рисунок 3.4 – Внутрішня модель блока *MOSFET*

Останній елемент у табл. 3.1 – ідеальний ключ *Ideal Switch* – не є моделлю конкретного напівпровідникового приладу. Він може бути використаний для спрощеного моделювання будь-якого керованого ключа, автоматичного вимикача або рубильника. Ключ повністю керується логічним сигналом g , незалежно від знака напруги між його затискачами, і є безінерційним, тобто миттєво переходить з одного стану в інший. Його початковий стан визначається параметром *Initial state*: 0 – розімкнутий, 1 – замкнутий. Цей блок вже розглядався при моделюванні перехідних процесів у колі постійного струму (див. п. 2.4, рис. 2.27, б).

Усі блоки в табл. 3.1 мають вимірювальний вихід m , який містить два сигнали – струм через прилад і напругу між його затискачами. Їх можна розділити за допомогою демультиплексора *Demux* і подати на окремі входи осцилографа *Scope*. Вихід m можна усунути з піктограми блока, знявши прапорець *Show measurement port* (див. рис. 3.2).

Крім моделей окремих напівпровідникових приладів, у бібліотеці *Power Electronics* є універсальний блок *Universal Bridge* (рис. 3.5), що моделює різні мостові схеми перетворювачів напруги. Його вікно налаштування показане на рис. 3.6. Кількість плечей моста (від 1 до 3) і тип силових ключів задаються в полях *Number of bridge arms* і *Power Electronic device* відповідно. Набір інших параметрів у вікні

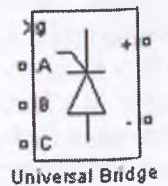


Рисунок 3.5 – Блок *Universal Bridge*

змінюється в залежності від того, з яких елементів складається схема (діодів, тиристорів, IGBT та ін.). Вікно на рис. 3.6 показано для триплечового моста, який складається із IGBT, шунтованих зворотними діодами. Параметри блока в цілому відповідають описанам вище для блоків *IGBT* і *Diode* (тут V_{fd} – гранична напруга діода). Конфігурація мостової схеми для цього випадку наведена на рис. 3.7.

У більш ранніх версіях MATLAB існує додатковий параметр налаштування блока *Universal Bridge* – *Port configuration*, у випадковому списку якого вибирається один із двох варіантів: *ABC as input terminals* або *ABC as output terminals*. У першому варіанті, встановленому за замовчуванням, порти А, В і С використовуються як вхідні, до яких підключається джерело синусоїдної напруги, а порти «+» і «-» – як вихід для підключення навантаження (постійна напруга). Тобто у цьому випадку блок використовується як *AC/DC*-конвертор (перетворювач змінної напруги в постійну). В другому варіанті – навпаки, порти міняються місцями, і живлення подається на входи «+» і «-», а навантаження підключається до трифазного виходу А, В, С. Блок перетворює постійну напругу в синусоїдну (*DC/AC*-конвертор). У розглядуваній версії MATLAB 2008b зазначений параметр відсутній, оскільки будь-який *SPS*-порт (маркований знаком □) може бути як вхідним, так і вихідним, у залежності від того, куди підключене джерело, а куди – навантаження. Тому тепер для зміни типу перетворювача досить вибрати потрібний тип джерела (*AC* чи *DC*) і підключити до відповідних затисків.

Number of bridge arms:	3
Snubber resistance R_s (Ohms)	1e5
Snubber capacitance C_s (F)	inf
Power Electronic device:	IGBT / Diodes
R_{on} (Ohms)	1e-3
Forward voltages [Device V_f (V), Diode V_{fd} (V)]	[0 0]
[T_f (s), T_t (s)]	[1e-6, 2e-6]
Measurements:	None

Рисунок 3.6 – Вікно налаштування блока *Universal Bridge*

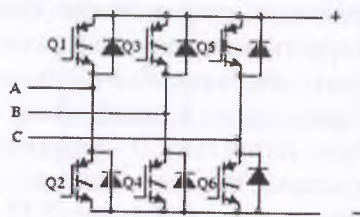


Рисунок 3.7 – Принципова схема блока *Universal Bridge* з елементами *IGBT/Diodes*

Вхід *g* на піктограмі блока *Universal Bridge* (див. рис. 3.5) призначений для керування перетворювачем. Даний блок може містити від 2 до 6 напівпровідникових елементів, тому, на відміну від розглянутих раніше елементарних блоків, для *Universal Bridge* сигнал на вході *g* має бути векторним і складатися з відповідної кількості сигналів керування. Моделювання систем керування буде розглянуто в п. 3.2.

Параметр *Measurement* у вікні налаштування блока призначений для виміру струмів і напруг моста за допомогою вимірника *Multimeter* (див. п. 2.2.5.1). Після відповідного налаштування даного параметра можуть бути виміряні наступні сигнали: напруги (*Device voltages*) або струми (*Device currents*) силових ключів, лінійні синусоїдні напруги (*UAB, UBC, UCA*) і постійна напруга (*UDC*), а також всі струми і напруги перетворювача (*All voltages and currents*).

3.1.2 Модель біполярного транзистора

У бібліотеці елементів *Power Electronics* відсутня модель біполярного транзистора (БП). Імовірно, це пов'язано з тим, що звичайні біполярні транзистори останнім часом витісняються з ринку більш перспективними пристроями *IGBT* і *MOSFET*. Тому в разі потреби модель БП необхідно скласти самостійно на основі відомих схем заміщення.

При роботі транзистора в області малих сигналів (на лінійній ділянці ВАХ) його можна представити у вигляді активного лінійного чотириполосника, якому відповідає схема заміщення на рис. 3.8, а (схема зі спільним емітером). Вона описується у вигляді системи рівнянь у *h*-параметрах [9, 13].

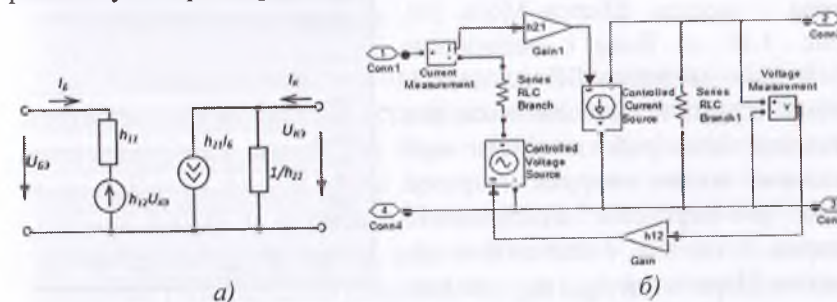


Рисунок 3.8 – Схема заміщення (а) і модель (б) біполярного транзистора в *h*-параметрах

Значення h -параметрів наводяться в довідкових даних на транзистори і мають наступний фізичний зміст: h_{11} і h_{21} – вхідний опір і коефіцієнт передачі струму емітера при короткому замиканні на виході транзистора; h_{12} і h_{22} – коефіцієнт зворотного зв'язку по напрузі та вихідна провідність при зворотному неробочому ході на вході транзистора. Перевагою системи в h -параметрах є відносна простота безпосереднього вимірювання значень коефіцієнтів h за допомогою дослідів неробочого ходу і короткого замикання. На практиці ця система використовується для аналізу низькочастотних схем.

Джерело струму $h_{21}I_B$ на рис. 3.8, а називається залежним, оскільки значення його струму залежить від струму іншої гілки – бази I_B . Аналогічно джерело напруги $h_{12}U_{KE}$ є залежним, оскільки характеризує зворотний зв'язок по вихідній напрузі «колектор-емітер» U_{KE} . На рис. 3.8, б наведена модель БП, яка відповідає еквівалентній схемі в h -параметрах. Залежні джерела моделюються за допомогою блоків *Controlled Current Source* і *Controlled Voltage Source*, розглянутих у п. 2.5.1. Сигнал керування на них надходить від відповідних вимірників струму і напруги через підсилювальні ланки *Gain* з коефіцієнтами h_{21} і h_{12} . Ця модель є підсистемою, яка може бути включена в іншу модель у вигляді окремого блока за допомогою вхідних і вихідних портів. Створення підсистем докладно розглядалося в п. 1.4. Для зручності задавання h -параметрів підсистема може бути оформлена у вигляді маски (див. п. 1.4.3) з власним вікном налаштування і піктограмою. Приблизний вигляд вікна налаштування маскованого блока *Transistor* показаний на рис. 3.9.

Інша розповсюджена схема заміщення біполярного транзистора – модель Еберса-Мола [6], рис. 3.10, а. Вона є нелінійною фізичною моделлю БП на постійному струмі і застосовується для аналізу його роботи тільки при великих змінах напруги і струму. Два p - n -переходи представлені двома зустрічно включеними діодами. Параметри α_{ON} і α_{OI} – це статичні коефіцієнти передачі струму відповідно при нормальному та

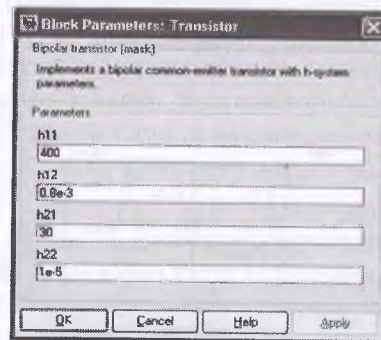


Рисунок 3.9 – Вікно налаштування моделі транзистора

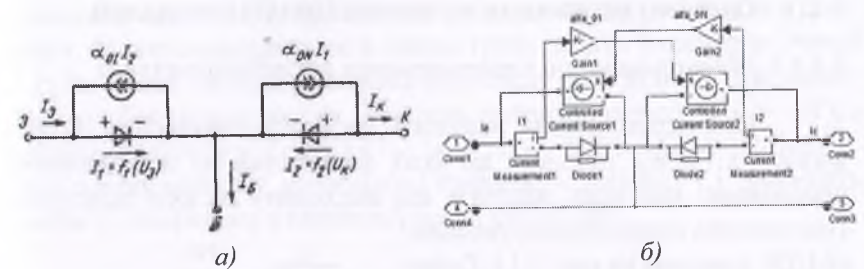


Рисунок 3.10 – Схема заміщення Еберса-Молла (а) і модель (б) біполярного транзистора

інверсному включенні транзистора. Залежні джерела струму $\alpha_{OI}I_2$ і $\alpha_{ON}I_1$ зображують струми збирання носіїв заряду, які залежать від струмів інжектіваних носіїв $p - I_1$ та I_2 . Модель, яка відповідає цій схемі заміщення БП зі спільною базою, зображена на рис. 3.10, б. Аналогічно до попередньої, вона також є підсистемою, яка може бути оформлена за допомогою редактора маски *Mask Editor* і включена в будь-яку модель.

Модель БП на рис. 3.10, б є спрощеною, оскільки вона побудована на основі спрощених діодів, вольтамперні характеристики яких складаються з двох лінійних ділянок (табл. 3.1). Для більш докладного моделювання можна скористатися математичним описом транзистора у вигляді аналітичних виразів, наведених у літературних джерелах, наприклад у [6, 13]. Ці рівняння являють собою нелінійні залежності $I_1 = f_1(U_E)$ і $I_2 = f_2(U_K)$. Принцип складання моделей на основі математичного опису був докладно розглянутий у п. 2.4.2.

3.2 Моделювання систем керування перетворювачами

У більшості випадків при моделюванні напівпровідникових перетворювачів як системи керування можна використовувати спеціалізовані блоки пакета *Sim Power Systems*, які знаходяться в бібліотеці *Extra Library/Control Blocks*. У тих випадках, коли жоден з готових блоків не підходить, можна скласти систему керування з функціональних елементів пакета *Simulink*. У даному параграфі розглянемо різні варіанти реалізації систем керування на прикладах моделей декількох найбільш розповсюджених перетворювальних пристроїв.

3.2.1 Системи керування на основі Simulink-моделей

3.2.1.1 Підвищувальний перетворювач постійної напруги

Цей перетворювач належить до класу імпульсних джерел живлення (ІДЖ), принцип дії яких заснований на періодичному перериванні постійної напруги, що надходить на вхід пристрою. Принципова схема підвищувального ІДЖ наведена на рис. 3.11. Основним елементом схеми є керований ключ Sw , у ролі якого використовується транзистор. Періодичне замикання і розмикання ключа приводить до стрибкоподібної зміни напруги дроселя і пульсації струму в ньому. В результаті на виході перетворювача маємо пульсуючу напругу, яка перевищує вхідну за значенням. Виходячи зі специфіки роботи пристрою, зрозуміло, що система керування повинна забезпечувати комутацію ключа з певною частотою, причому співвідношення між тривалістю замкнутого і розімкнутого станів обумовлює значення напруги на виході.

Модель підвищувального джерела з транзистором IGBT зображена на рис. 3.12. В даному випадку розглядається найбільш простий варіант перетворювача, тобто коли немає необхідності забезпечувати стабілізацію напруги на виході, а потрібно просто підвищити вхідну напругу джерела з 60 до 100 В. У такому варіанті у ролі системи керування зручніше використовувати блок *Pulse*

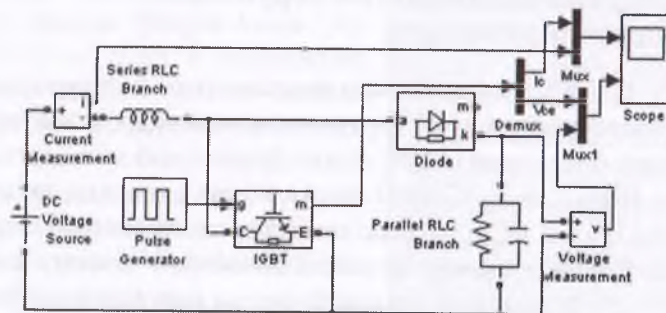


Рисунок 3.12 – Модель підвищувального ІДЖ

Generator (див. п. 1.2.1), що генерує імпульси заданої частоти і ширини. У вікні налаштування блока треба задати параметри *Period* і *Pulse Width* (період і ширина імпульсів відповідно). При цьому частота f обчислюється як величина, зворотна періоду T ($f = 1/T$), а ширина імпульсу γ (скважність) задається у відсотках від періоду. У загальному випадку, коли відомі тривалість імпульсу t_i і тривалість паузи t_n , коефіцієнт γ визначається за формулою

$$\gamma = \frac{t_i}{t_i + t_n} = \frac{t_i}{T}. \quad (3.1)$$

Для підвищувального імпульсного джерела значення γ , необхідне для одержання вихідної напруги U_{out} при заданій вхідній U_{in} , розраховується за виразом

$$\gamma = 1 - \frac{U_{in}}{U_{out}}. \quad (3.2)$$

У даному випадку $\gamma = 0,4$, тобто 40% тривалості всього періоду транзистор перебуває в провідному стані. Параметр *Phase delay* забезпечує затримку початку генерування імпульсів на задану кількість секунд (для даної моделі встановлений у 0). Результати роботи моделі показані на рис. 3.13. Жирними лініями позначений струм дроселя i_L і напруга на навантаженні U_{out} , а тонкими лініями – i_{CE} та U_{CE} – струм колектора і напруга колектор-емітер – транзистора відповідно. Струм і напруга діода також можуть бути виміряні через вихід *m*. Наведені графіки показують, що вихідна напруга U_{out} становить 100 В при заданій вхідній напрузі $U_{in} = 60$ В.

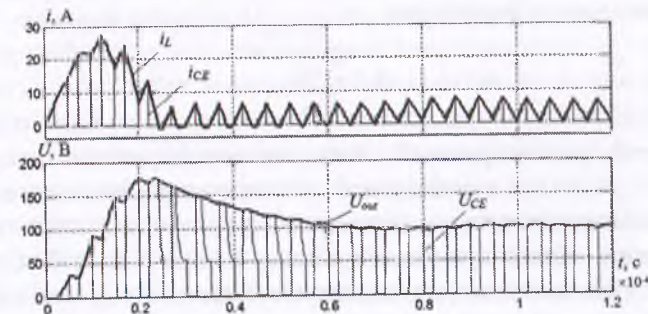


Рисунок 3.13 – Осцилограми в підвищувальному ІДЖ

3.2.1.2 Знижувальний перетворювач постійної напруги

Розглянутий блок *Pulse Generator* є прикладом найпростішої системи керування, оскільки в ньому не передбачена зміна параметрів у процесі сеансу моделювання. Іншими словами, параметри імпульсів (амплітуда, період, ширина) повинні бути задані заздалегідь у вікні налаштування і не можуть бути змінені, поки не завершиться розрахунок моделі. Однак може виникнути необхідність моделювання перетворювача, в якому система керування змінює параметр γ (*Pulse Width*), а отже, і вихідну напругу, безпосередньо під час розрахунку моделі. Наприклад, імпульсний стабілізатор напруги, в якому вихідний сигнал підтримується на заданому рівні незалежно від коливань вхідної напруги.

Для ілюстрації даного прикладу розглянемо знижувач ІДЖ зі стабілізацією напруги (рис. 3.14). Воно складається з тих же елементів, що і підвищувальний перетворювач, але в даній схемі керований ключ встановлено на вході пристрою. В моменти замкнутих станів ключа дросель L накопичує енергію, а при розмиканні ключа – розряджається за колом «діод-навантаження». У результаті напруга на навантаженні стає меншою за вхідну і може регулюватися зміною параметра γ , тобто співвідношенням (3.1) між тривалостями замкнутого і розімкнутого станів ключа. Для знижувача ІДЖ при заданих напругах на вході і виході схеми коефіцієнт γ визначається за формулою

$$\gamma = \frac{U_{out}}{U_{in}}. \quad (3.3)$$

Модель перетворювача і його система керування наведені на рис. 3.15, а і 3.15, б відповідно. Розглянемо випадок, коли в процесі моделювання напруга на вході кілька разів змінює своє значення. Для цього в моделі замість звичайного джерела постійної напруги *DC Voltage Source* використане кероване джерело *Controlled Voltage Source*, напруга якого задається за допомогою блока *Timer*.

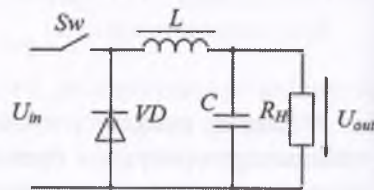
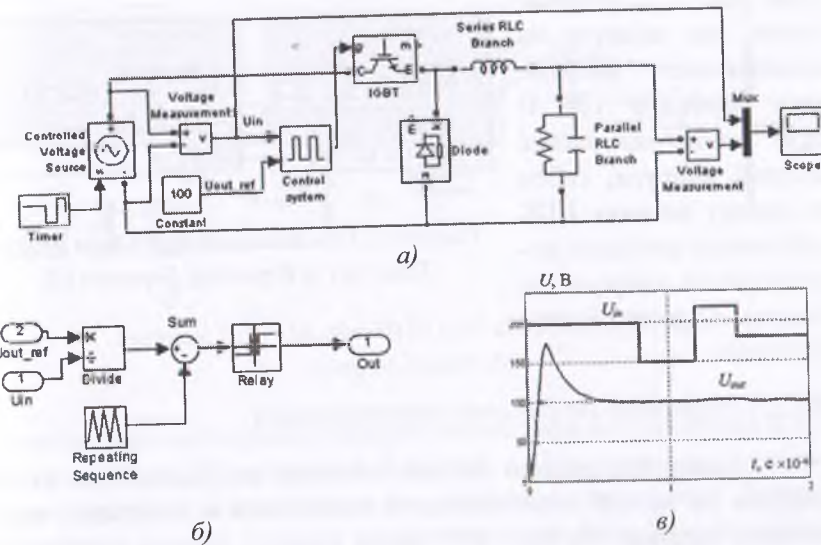


Рисунок 3.14 – Принципова схема знижувального ІДЖ

Рисунок 3.15 – Модель знижувального ІДЖ (а); модель системи управління *Control System* (б); осцилограми напруги перетворювача (в)

У ролі ключа, як і в попередньому випадку, застосовується блок *IGBT*. Очевидно, що оскільки напруга на вході кола змінюється, то для постійності напруги на навантаженні параметр γ теж повинен змінити своє значення відповідно до (3.3). Це забезпечує система керування, виконана у вигляді підсистеми *Control system*, піктограма якої для наочності оформлена за допомогою редактора маски *Mask Editor* (див. п. 1.4.3). Дана система керування побудована за так званим вертикальним принципом. На її вхід надходить сигнал у вигляді значення напруги $U_{out-ref}$, яке потрібно підтримувати на виході перетворювача, а також виміряна вхідна напруга U_{in} .

Блок *Divide* формує відношення цих двох сигналів (див. рис. 3.15, б), і отриманий у результаті керуючий сигнал зі значенням γ порівнюється з пилкоподібною напругою блока *Repeating Sequence*. В моменти рівності керуючого і пилкоподібного сигналів блок *Relay* формує імпульс, що надходить на вхід g транзистора. Вікна налаштування блоків *Timer* і *Repeating Sequence* наведені на рис. 3.16. У верхніх полях обох вікон вводяться значення часу, а в нижніх – відповідні значення вихідного сигналу. Налаштування блока *Relay* лишаються заданими за замовчуванням. Результати моделювання

(див. рис. 3.15, в) показують, що напруга на навантаженні зберігає своє значення 100 В при різних значеннях вхідної напруги, тобто в даному випадку ІДЖ забезпечує не тільки перетворення напруги за рівнем, а також і її стабілізацію.

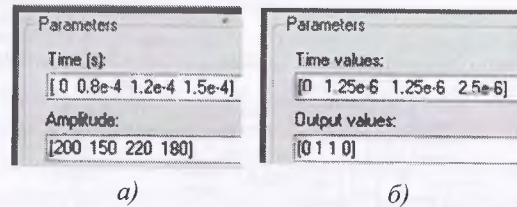


Рисунок 3.16 – Вікна налаштування блоків *Timer* (а) та *Repeating Sequence* (б)

3.2.1.3 Широтно-імпульсний перетворювач

Тепер розглянемо варіант системи керування, при якому напруга на виході перетворювача змінюється в результаті зміни сигналу задання. На рис. 3.17 наведена принципова схема широтно-імпульсного перетворювача напруги (ШІП), що найчастіше використовується в електроприводах постійного струму. Перетворювач складається з чотирьох транзисторних ключів зі зворотними діодами, навантаження включене у діагональ моста. При симетричному способі керування ключі працюють попарно – VT1, VT4 і VT2, VT3. При цьому напруга на навантаженні являє собою двополярні (знакозмінні) імпульси, а середнє значення цієї напруги визначається скважністю імпульсів γ . При несиметричному керуванні переключаються тільки транзисторні ключі VT3 і VT4, ключ VT1 постійно відкритий, а ключ VT2 постійно закритий. У цьому випадку напруга на навантаженні однополярна, що дозволяє зменшити пульсації струму навантаження.

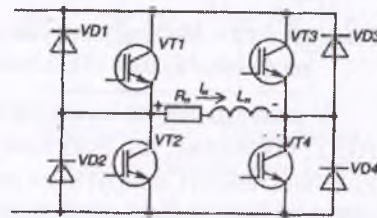


Рисунок 3.17 – Принципова схема ШІП

Модель ШІП при роботі на активно-індуктивне навантаження наведена на рис. 3.18, а. Міст моделюється блоком *Universal Bridge* із силовими модулями *IGBT/Diodes*. Кількість плечей моста дорівнює 2 (*Number of bridge arms*). Нехай у процесі моделювання

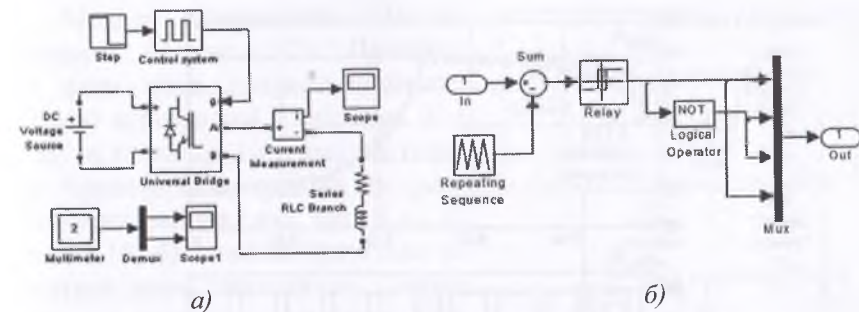


Рисунок 3.18 – Моделі ШІП (а) і системи керування *Control System* (б)

потрібно здійснити реверс напруги навантаження зі значення $0,5U_{\text{нх}}$ до значення $-0,5U_{\text{нх}}$ в момент часу 0,05 с. При цьому на вході системи керування *Control system* має змінитися коефіцієнт γ . Джерело керуючого сигналу γ краще використовувати блок *Step*, параметри якого встановлені в наступні значення:

Step time: 0.05, *Initial value*: 0.5, *Final value*: -0.5.

Якби не було потрібно змінити напругу навантаження в процесі моделювання, то для реалізації симетричної системи керування досить було б використати два блоки *Pulse Generator* (по одному на кожну пару ключів) з відповідним значенням параметра γ , а для несиметричної системи – ще додатково два блоки *Constant* – для подачі постійних сигналів на ключі, які не змінюють свого стану.

Внутрішня модель системи керування *Control system* для симетричного закону керування ШІП наведена на рис. 3.18, б. Вона подібна до розглянутої раніше моделі для знижувального перетворювача (3.15, б) з тією лише різницею, що в даному випадку значення γ не розраховується в підсистемі, а надходить на її вхід із блока *Step*.

Крім того, оскільки міст має 4 ключі, то підсистема генерує 4 імпульси. Блок *Logical Operator* тут реалізує інверсію сигналу, щоб дві пари ключів працювали в протифазі. В налаштуваннях блока *Repeating Sequence* параметр *Time values* встановлений у значення [0 0,001 0,001 0,002], а параметр *Output values* – у значення [-1 1 1 -1]. Осцилограми струму $i_{\text{н}}$ і напруги $u_{\text{н}}$ на навантаженні показані на рис. 3.19.

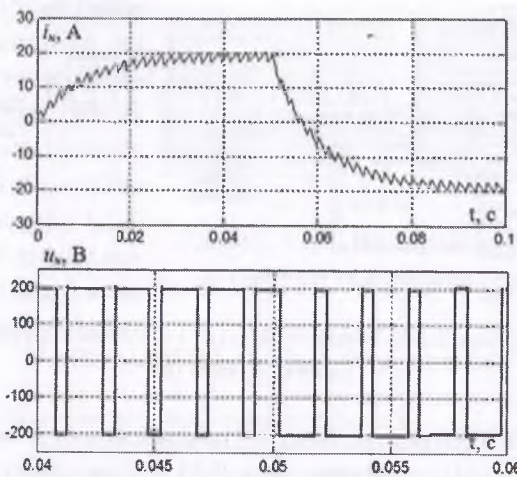


Рисунок 3.19 – Графіки перехідних процесів у ШП

3.2.1.4 Однофазний керований випрямляч із середньою точкою

Розглянемо приклад системи керування, в якій керуючий імпульс синхронізований з напругою мережі і зсунутий на кут α відносно початку півхвилі напруги живлення. Як правило, такі системи застосовуються для керування перетворювачами, побудованими на основі тиристорів. На рис. 3.20 наведена принципова схема однофазного двопівперіодного керованого випрямляча (КВ) з середньою точкою, який працює на активно-індуктивне навантаження. Вторинна обмотка трансформатора має дві півобмотки зі спільним (нульовим) виводом, їхні напруги U_{21} і U_{22} однакові та протилежні за фазою. Отже, два вентиля працюють по черзі – у залежності від того, на якому з них напруга додатна в кожен момент часу. Для регулювання вихідної напруги змінюють кут комутації α (момент відкриття) тиристорів за допомогою системи керування, яка періодично, синхронно із напругою мережі, генерує керуючі імпульси. У залежності від тривалості провідного стану тиристорів змінюється середнє значення напруги на навантаженні.

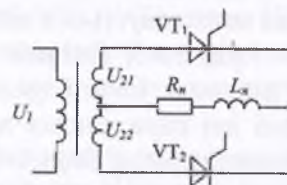


Рисунок 3.20 – Принципова схема однофазного КВ

Модель однофазного КВ зображена на рис. 3.21. Початкові фази обох джерел (параметр *Phase*) встановлені в значення 0° і 180° відповідно. Система керування тиристорами подібна до розглянутих раніше (див. рис. 3.15, б, рис. 3.18, б) і також будується за вертикальним принципом. Сигнал *alfa*, що відповідає значенню кута керування α (у градусах), порівнюється з пилоподібним сигналом, і в момент їхньої рівності формується імпульс. Пилоподібна напруга тут є синхронізуючим сигналом, частота якого вдвічі більша за частоту напруги живлення. Тривалість імпульсу 10° визначається блоком *Switch* (п. 1.2.2.1), параметр *Threshold* якого встановлений у значення 10. У даній моделі імпульси виробляються кожний півперіод і подаються на обидва блоки *Thyristor* одночасно, але в кожен момент часу відкритий буде саме той тиристор, напруга на якому додатна, незалежно від керуючого сигналу на вході *g*. При необхідності можна реалізувати розподіл імпульсів по двох каналах керування, щоб наблизити модель до реальної системи. Осцилограми струму i_n і напруги u_n навантаження при $\alpha = 60^\circ$ показані на рис. 3.22.

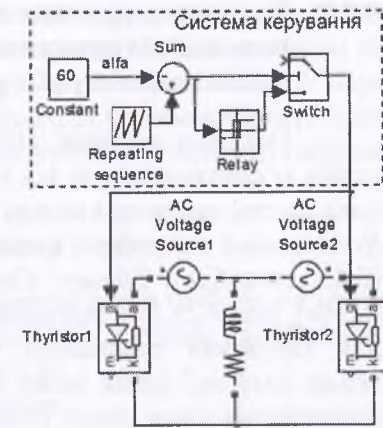


Рисунок 3.21 – Модель однофазного КВ

У вікні налаштування блока *Repeating Sequence* вводяться наступні значення параметрів (через пробіл): *Time values* – $[0 \ 1/(2 \cdot f)]$; *Output values* – $[0 \ 180]$. Тут $f = 60$ Гц – частота вхідної напруги, що має відповідати частоті, заданій у налаштуваннях синусоїдних джерел. Налаштування блока *Relay* залишаються заданими за замовчуванням.

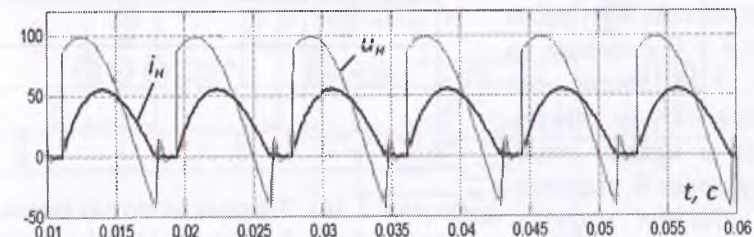


Рисунок 3.22 – Осцилограми в однофазному КВ

3.2.2 Системи керування на основі генератора імпульсів із синхронізацією від мережі *Synchronized 6-pulse generator*

Оскільки керовані випрямлячі відносяться до класу пристроїв із синхронізацією від напруги мережі, то при моделюванні їхніх систем керування можна використовувати спеціальний блок – *Synchronized 6-pulse generator* (бібліотека *Extra Library/ Control Blocks*), рис. 3.23. Він призначений для керування трифазною мостовою схемою, однак може бути адаптований і для інших різновидів тиристорних схем. Цей блок являє собою генератор шести послідовностей імпульсів, зсунутих на кут α відносно початку півхвилі лінійної напруги і на кут 60° відносно один одного. Перший вхід *alpha-deg* призначений для задання кута комутації α (у градусах), на наступні три входи *AB*, *BC*, *CA* подаються миттєві значення лінійних синхронізуючих напруг трифазної мережі, останній вхід *Block* призначений для блокування генератора: якщо сигнал на цьому вході більше нуля, то подача імпульсів припиняється.

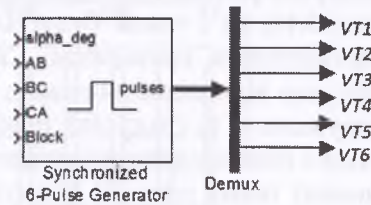


Рисунок 3.23 – Модель генератора імпульсів

Вихід *pulses* являє собою векторний сигнал, який складається з 6 окремих сигналів (по одному на кожен тиристор). Діаграма розподілу імпульсів по всіх каналах керування при $\alpha = 0$ показана на рис. 3.24. Видно, що кожен сигнал генерується в момент переходу через 0 відповідної лінійної напруги. Передбачений також

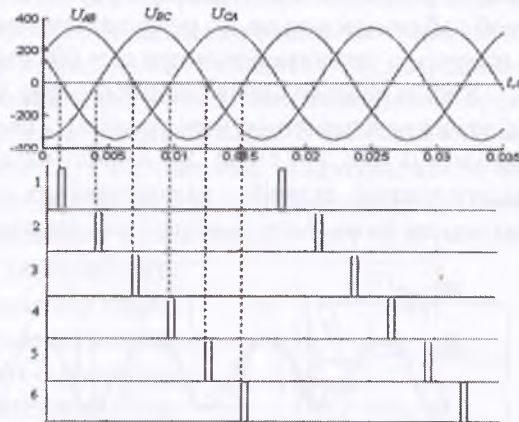


Рисунок 3.24 – Діаграма розподілу імпульсів на виході блока *Synchronized 6-pulse generator*

режим подвійної пульсації (встановлений прапорець *Double pulsing* у вікні налаштування блоку), коли на кожному каналі протягом періоду формується не один, а два імпульси з інтервалом 60° . У параметрах налаштування генератора задаються значення частоти синхронізуючої напруги (*Frequency of synchronization voltages*) і ширина керуючого імпульсу (*Pulse width*).

3.2.2.1 Трифазний мостовий керований випрямляч

На рис. 3.25 зображена принципова схема трифазного мостового випрямляча, який працює на активно-індуктивне навантаження і проти-ЕРС, а його модель – на рис. 3.26. Трифазне реальне джерело напруги моделюється трьома однофазними ідеальними джерелами, послідовно з якими включені активно-індуктивні опори.

Початкові фази кожного джерела встановлені в значення 0, -120° і 120° відповідно. Синхронізуючі лінійні напруги U_{AB} , U_{BC} , U_{CA} вимірюються блоками *Voltage Measurement* і подаються на відповідні входи генератора імпульсів.

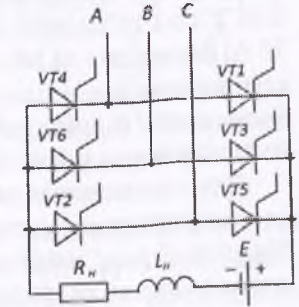


Рисунок 3.25 – Схема трифазного мостового КВ

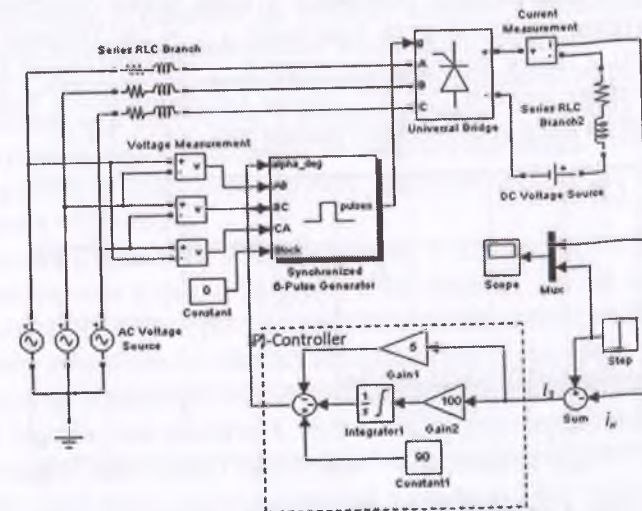


Рисунок 3.26 – Модель трифазного мостового КВ

Якщо міст складається з окремих блоків *Thyristor* чи *Detailed Thyristor*, то векторний вихід генератора треба розділити на 6 окремих сигналів, як показано на рис. 3.23, і подати ці сигнали на керуючі входи тиристорів відповідно до їх нумерації на рис. 3.25.

Якщо ж мостом є блок *Universal Bridge*, то в цьому випадку розділяти сигнали немає необхідності, і вихід генератора імпульсів підключається безпосередньо на вхід *g* мостового блока. Система керування реалізована на основі ПІ-регулятора струму, на вхід якого надходить різниця сигналів задання I_3 (блок *Step*) і зворотного зв'язку (струм i_n , вимірюваний у колі навантаження).

Результат моделювання показаний на рис. 3.27. У момент часу 0,04 с сигнал задання I_3 стрибком змінює своє значення з 10 до 30 А. Відповідно на виході ПІ-регулятора змінюється значення кута керування α , що надходить на вхід *alpha-deg* генератора імпульсів *Synchronized 6-pulse generator*. У результаті струм навантаження i_n на виході моста також змінюється відповідно до задання.

Для моделювання регуляторів різних типів (П-, ПІ-, ПІД-) можна використовувати спеціальний блок *PID Controller* (бібліотека *Simulink Extras/ Additional Linear*), у налаштуваннях якого задаються значення коефіцієнтів пропорційної, інтегральної і диференціальної складових (п. 1.3.1.7). Однак у даному випадку регулятор складений з окремих ланок, оскільки потрібне додаткове налаштування його інтегральної складової: у вікні блока *Integrator1* задане початкове значення кута комутації $\alpha_{\text{поч}} = 20^\circ$ (параметр *Initial condition*), а також його верхня межа $\alpha_{\text{max}} = 160^\circ$ (*Upper saturation limit*) і нижня межа $\alpha_{\text{min}} = 20^\circ$ (*Lower saturation limit*). Опис блока *Integrator* та його вікно налаштування див. у п. 1.3.1.6.

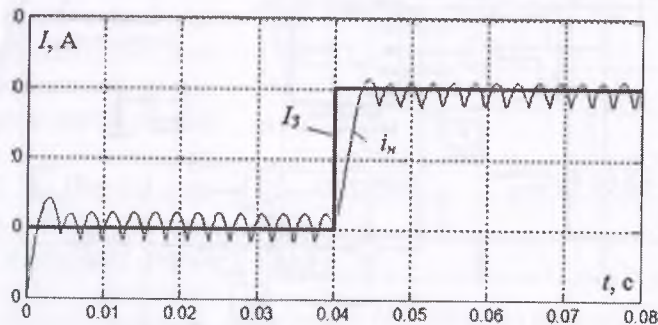


Рисунок 3.27 – Графіки переходних процесів у мостовому трифазному КВ

3.2.2.2 Тиристорний перетворювач напруги

Тепер розглянемо застосування генератора імпульсів *Synchronized 6-pulse generator* у моделях, які потрібно складати з окремих тиристорів, без використання універсального мостового блока. Прикладом такого пристрою може служити тиристорний перетворювач напруги (ТПН), який застосовується для керування асинхронними двигунами, нагрівальними установками тощо.

Трифазний ТПН являє собою три пари зустрічно-паралельно включених тиристорів у кожній фазі (рис. 3.28). Регулювання напруги здійснюється зміною кута відкриття тиристорів α у межах кожного окремого півперіоду напруги мережі. Тим самим змінюється форма напруги і, отже, її діюче значення.

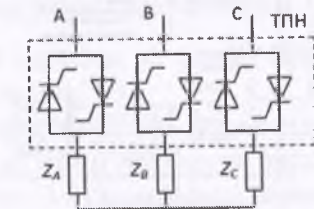


Рисунок 3.28 – Принципова схема ТПН

Регулювання напруги від максимального значення до нуля при активному навантаженні відбувається при зміні кута комутації α у межах $0^\circ < \alpha < 150^\circ$, а при активно-індуктивному навантаженні $\varphi < \alpha < 150^\circ$, де φ – кут навантаження.

На рис. 3.29, а показана модель трифазного кола з активним навантаженням, з'єднаним за схемою «зірка» з нульовим проводом. Тиристорний перетворювач виконаний у вигляді підсистеми *Thyristor Converter*, внутрішня модель якої зображена на рис. 3.29, б, і включений між джерелом і навантаженням. У даному випадку для керування перетворювачем за допомогою блока *Synchronized 6-pulse generator* доводиться вручну реалізувати розподіл імпульсів по окремих тиристорах.

Оскільки тиристори ТПН включені в кожну фазу мережі, то синхронізуючою є фазна напруга, а не лінійна, як для мостового випрямляча. Напруги трьох джерел вимірюються блоком *Multimeter* (параметр *Measurement* джерел встановлений у значення *Voltage*) і надходять на відповідні входи генератора імпульсів. Вихідний векторний сигнал, що містить 6 імпульсів, керує тиристорами ТПН. Для того щоб розподілити імпульси по тиристорах, необхідно знати, який з імпульсів синхронізований з кожною з фаз напруги мережі. Це можна визначити за допомогою діаграми на рис. 3.24.

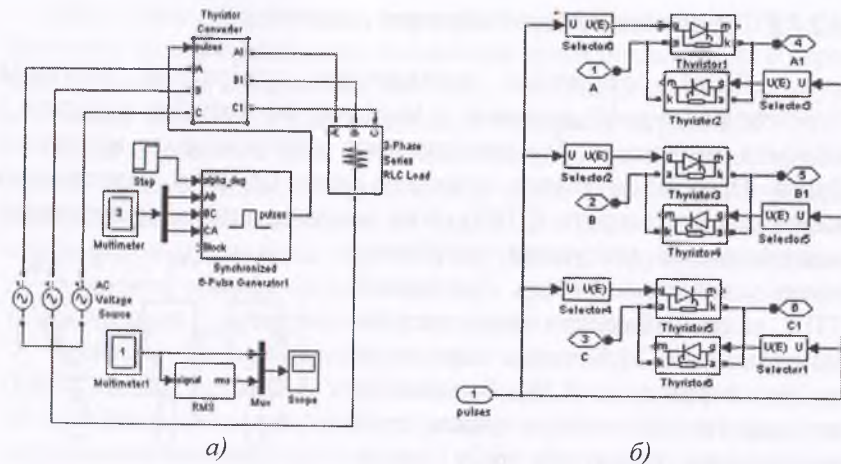


Рисунок 3.29 – Модель трифазного кола з ТПН (а);
модель ТПН Thyristor Converter (б)

З діаграми видно, що початку додатної півхвилі напруги U_{AB} відповідає імпульс 6-го каналу, а початку від'ємної півхвилі U_{AB} – імпульс 3-го каналу. Таким чином, якщо на вхід AB генератора подається напруга фази A джерела, то тиристори, включені в цю фазу, мають керуватися імпульсами 6 і 3. Аналогічно визначаємо, що тиристорам фази B відповідають імпульси 2 і 5, а тиристорам фази C – імпульси 4 і 1. Для розподілу імпульсів можна використовувати блок *Selector* (бібліотека *Simulink/Signal Routing*), п. 1.2.2.3. Цей блок із вхідного векторного сигналу розмірності n (у даному випадку $n = 6$) виділяє один або кілька сигналів і пропускає їх на вихід, а інші сигнали блокує. В моделі ТПН на рис. 3.29, б у налаштуваннях кожного блока *Selector* указаний номер відповідного імпульсу (поле *Elements*), а також розмірність вхідного сигналу n (поле *Input port width*). Кут комутації α формується блоком *Step* і надходить на вхід *alpha-deg* генератора. Значення α стрибком змінюється з 60 до 120° у момент часу 0,1 с.

На рис. 3.30 показані осцилограми напруг у колі: u_n – напруга фази A на навантаженні, U_n – її діюче значення (вимірюється блоком *RMS*), u_A – напруга фази A джерела. Видно, що при зміні кута комутації змінюється форма напруги і, отже, її діюче значення.

За допомогою генератора імпульсів можна реалізувати систему керування для моделі однофазного випрямляча, розглянутого

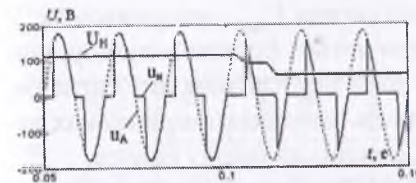


Рисунок 3.30 – Осцилограми напруг у трифазному колі з ТПН

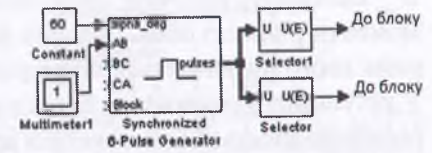


Рисунок 3.31 – Модель системи керування однофазним КВ

в п. 3.2.1.4 на рис. 3.21. Модель такої системи на основі блока *Synchronized 6-pulse generator* наведена на рис. 3.31.

Оскільки коло однофазне, то використовується тільки один синхронізуючий вхід генератора U_{AB} – на нього надходить напруга джерела *AC Voltage Source1* (див. рис. 3.21), виміряна блоком *Multimeter*. На виході генератора два блоки *Selector* виділяють із спільного вектора сигнали 6-го і 3-го каналів керування і подають їх на керуючі входи тиристорів *Thyristor1* і *Thyristor2* відповідно. Ця модель системи керування зручніша, ніж зображена на рис. 3.21, оскільки вона не потребує зміни параметрів налаштування при зміні частоти синхронізуючого сигналу.

3.2.3 Системи керування на основі ШІМ-генератора PWM Generator

У системах автоматизованого електропривода силові регулятори широко використовуються автономні інвертори, призначені для перетворення постійного струму в змінний з регульованою частотою і напругою. Для керування силовими ключами інвертора існує багато різних способів. Останнім часом у зв'язку з досягненнями в галузі виготовлення потужних швидкодіючих транзисторів *IGBT* і *MOSFET* значне розповсюдження одержав спосіб керування за допомогою широтно-імпульсної модуляції (ШІМ). Він дає змогу значно поліпшити гармонічний склад напруги на виході інвертора у порівнянні з іншими способами керування. На рис. 3.32 наведені осцилограми, що пояснюють принцип синусоїдної ШІМ. Пилкоподібний сигнал високої частоти

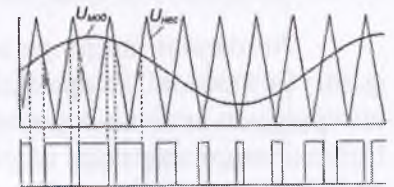


Рисунок 3.32 – Принцип ШІМ

$U_{\text{нес}}$ називають несучим, синусоїдний сигнал $U_{\text{мод}}$ – модулюючим. У моменти рівності обох сигналів формуються фронти імпульсу, ширина якого визначається шириною зони перетинання цих сигналів. У результаті утворюється послідовність імпульсів, ширина яких періодично зростає і зменшується відповідно до закону синуса.

У бібліотеці *Extra Library/Control Blocks* існує спеціальний блок – *PWM Generator* (рис. 3.33), призначений для моделювання систем керування із ШІМ–перетворювачами (*PWM – pulse width modulation*), побудованими на основі мостової транзисторної схеми.

Параметр *Generator Mode* вікна налаштування блока (рис. 3.34) визначає кількість плечей моста і, відповідно, кількість імпульсних сигналів (2, 4, 6 чи 12) на виході генератора. У полі *Carrier Frequency* вводиться частота несучого сигналу. Чим вона вище, тим краще гармонічний склад вихідної напруги. Модулюючий сигнал може надходити на вхід *Signal* (якщо знято прапорець *Internal generation of modulation signal*), або генеруватися всередині нього (прапорець установлений). В останньому випадку в додаткових полях задаються параметри модулюючого сигналу – коефіцієнт модуляції (відношення амплітуд модулюючої та пилоподібної напруг), частота і початкова фаза. Якщо прапорець знятий, то вхід *Signal* лишається непідключеним.

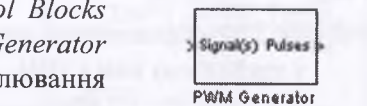


Рисунок 3.33 – Блок *PWM Generator*

Generator Mode	3-arm bridge (6 pulses)
Carrier frequency (Hz):	1080
<input checked="" type="checkbox"/> Internal generation of modulating signal(s)	
Modulation index (0<m<1):	0.4
Frequency of output voltage (Hz)	60
Phase of output voltage (degrees)	0

Рисунок 3.34 – Вікно налаштування блока *PWM Generator*

3.2.3.1 Автономний інвертор напруги

Автономні інвертори напруги (АІН) поділяються на однофазні і багатофазні. Як приклад розглянемо трифазний АІН із ШІМ, що працює на активно-індуктивне навантаження, принципова схема і модель якого зображені на рис. 3.35. У даному випадку кількість плечей мостового блока *Universal Bridge* дорівнює 3, а як силові ключі використовуються модулі *MOSFET/Diodes*. Відповідно, пара-

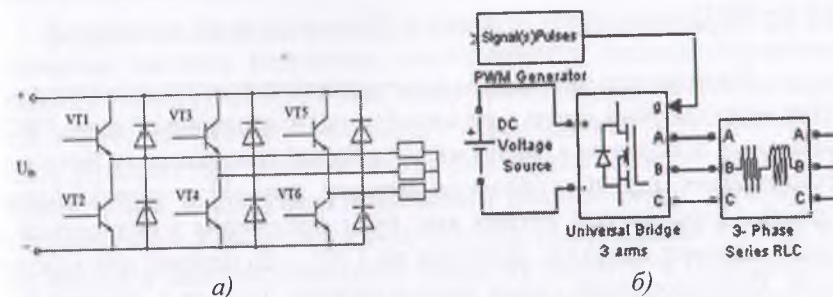


Рисунок 3.35 – Принципова схема (а) і модель (б) трифазного АІН з ШІМ

метр *Generator Mode* генератора імпульсів *PWM Generator* встановлений у значення *3-arm bridge (6 pulses)*.

Інші його параметри залишені заданими за замовчуванням. Модулюючий сигнал генерується всередині блока, тому вхід блока не підключений. Параметри цього сигналу (амплітуда і частота) визначають синусоїдну напругу на виході інвертора. На рис. 3.36 наведені осцилограми напруги і струму однієї фази навантаження, що вимірюються за допомогою блока *Multimeter* (на рис. 3.35 не показаний).

Видно, що форми сигналів близькі до синусоїдних. Таким чином, автономний інвертор можна вважати регульованим джерелом синусоїдної напруги.

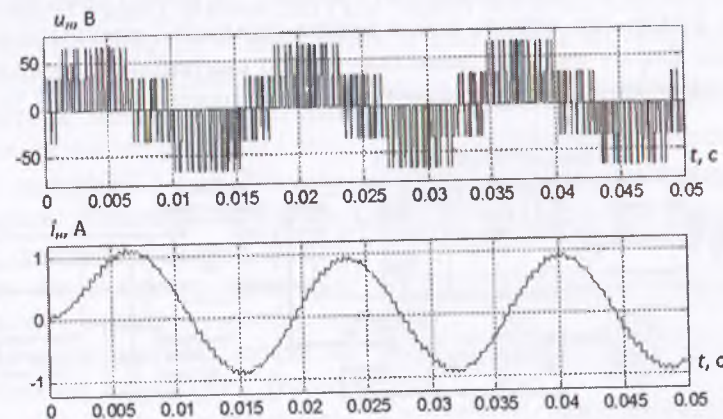


Рисунок 3.36 – Графіки перехідних процесів у трифазному АІН

3.2.3.2 Перетворювач частоти з ланкою постійного струму

На прикладі частотного перетворювача розглянемо випадок, коли модулюючий сигнал не виробляється всередині блока *PWM Generator*, а подається ззовні на вхід *Signal* генератора (у його налаштуваннях знятий прапорець *Internal generation of modulation signal*). Причому цей сигнал має бути векторним і складатися з трьох окремих сигналів, зсунутих на 120° – по одному для кожної фази. Функціональна схема перетворювача частоти з ланкою постійного струму наведена на рис. 3.37.

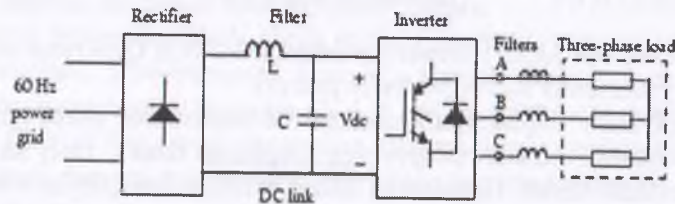


Рисунок 3.37 – Функціональна схема перетворювача частоти

На його вхід від трифазного джерела подається синусоїдна напруга з частотою 60 Гц, яка перетворюється в постійну за допомогою діодного моста (некерованого випрямляча). Випрямлена напруга надходить на LC-фільтр, і після згладжування пульсацій знову перетворюється автономним інвертором у змінну напругу, але вже з іншими значеннями амплітуди і частоти. На виході інвертора є ще один фільтр, що знижує вплив вищих гармонік і максимально наближає форму напруги навантаження до синусоїдної. Модель, що відповідає даній функціональній схемі, зображена на рис. 3.38.

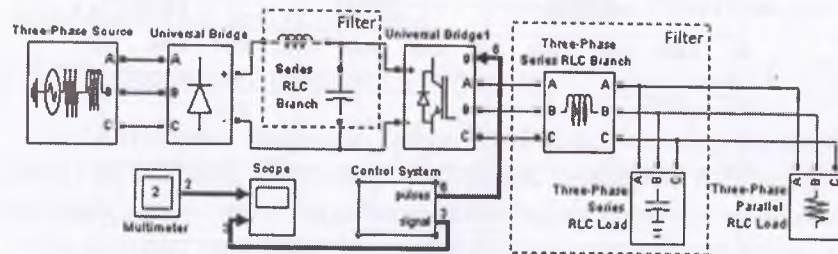
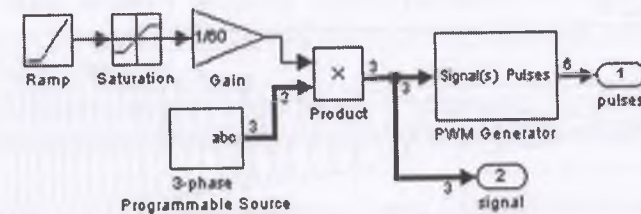


Рисунок 3.38 – Модель перетворювача частоти

Керування інвертором відбувається за принципом ШІМ, за допомогою системи керування, реалізованої у вигляді підсистеми *Control System* (рис. 3.39). Для генерування модулюючого сигналу, що керує генератором імпульсів *PWM Generator*, застосовується блок *3-phase Programmable Source* (бібліотека *Extra Library/Control Blocks*). Він аналогічний описаному раніше блоку *Three-Phase Programmable Voltage Source* (п. 2.3.1.2), різниця лише в тому, що на виході у використовуваного блока не напруга, а безрозмірний *Simulink*-сигнал, що дозволяє застосовувати його для реалізації систем керування.

Рисунок 3.39 – Модель системи керування перетворювачем частоти (підсистема *Control System*)

Вікно налаштування програмованого джерела показане на рис. 3.40. Параметри встановлені відповідно до заданого режиму: забезпечити плавне зростання частоти за час 0.3 с від 0 до 60 Гц. Кінцеве значення частоти забезпечується параметром *Rate of change* (швидкість зміни), рівним 200 Гц/с. Нехай потрібно реалізувати закон регулювання $U/f = \text{const}$, тобто зі зростанням частоти повинна пропорційно підвищуватися амплітуда напруги. Початкові значення амплітуди, фази і частоти задаються в полі *Positive-sequence*. Оскільки програмоване джерело дозволяє задати зміну тільки одного параметра (у даному випадку, частоти), то зміна амплітуди реалізована за допомогою додаткових блоків (див. рис. 3.39). Зростання частоти задається за допомогою

Positive-sequence: [Amplitude Phase(degrees) Freq.]
[1 0 0]
Time variation of: Frequency
Type of variation: Ramp
Rate of change (value/s): 200
Variation timing (s): [Start End]
[0 0.3]
<input type="checkbox"/> Harmonic generation:

Рисунок 3.40 – Вікно налаштування блока *3-phase Programmable Source*

блока *Ramp* (параметр $Slope = 200$), обмежується на рівні 60 Гц блоком *Saturation* ($Upper\ limit = 60$, $Lower\ limit = 0$), і далі за допомогою пропорційної ланки *Gain* за значенням частоти розраховується амплітуда сигналу відповідно до заданого закону регулювання. Блок *Product* здійснює множення вихідного сигналу джерела на отримане значення амплітуди. В результаті на вхід ШІМ-генератора надходить трифазний синусоїдний сигнал керування U_{contr} зі змінюваною в часі амплітудою від 0 до 1 і частотою від 0 до 60 Гц (рис. 3.41, а). Осцилограми напруги і струму однієї фази активного навантаження показані на рис. 3.41, б.

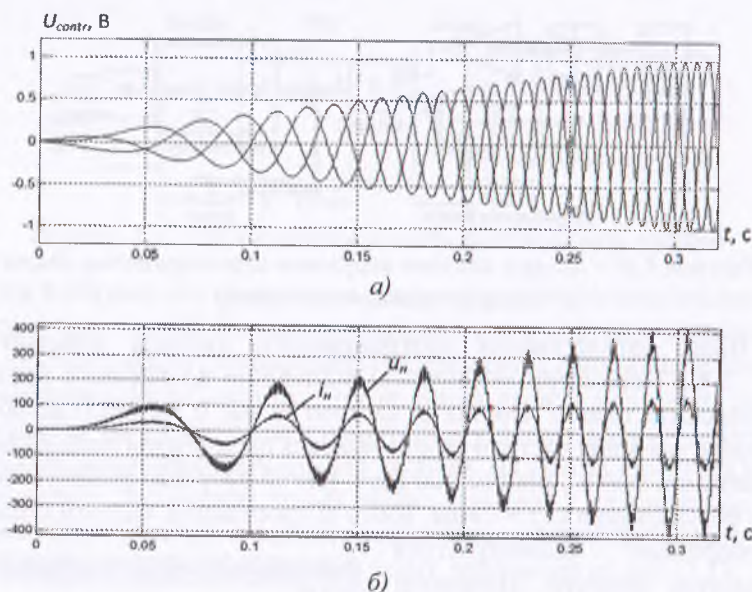


Рисунок 3.41 – Сигнал керування на вході блока *PWM Generator* (а); осцилограми струму і напруги на виході ПЧ (б)

Контрольні питання до розділу 3

1. Як відбувається керування моделями напівпровідникових елементів бібліотеки *Power Electronics*?
2. Чим відрізняються моделі тиристорів *Thyristor* і *Detailed Thyristor*?
3. Пояснити алгоритм роботи блока *MOSFET*. Дати його вольтамперну характеристику.
4. Як відбувається процес відключення блоків *IGBT* і *GTO*? Пояснити за допомогою діаграми струму.
5. Яким чином у моделях підвищувального та знижувального ІДЖ (див. рис. 3.12 і 3.15) відбувається регулювання значення вихідної напруги?
6. Пояснити призначення блока *Synchronized 6-pulse generator*. Які сигнали надходять на його входи? Який сигнал знімається з виходу?
7. Для чого призначений вхід *Signal* блока *PWM Generator*? У якому випадку даний вхід можна не використовувати?
8. Пояснити, яким чином у моделі перетворювача частоти (див. рис. 3.38) реалізовано закон керування $U/f = \text{const}$?

Завдання для самостійного виконання

1. Скласти модель підвищувального ІДЖ (див. рис. 3.12). Встановити наступні значення параметрів: $U_m = 100$ В; $L = 0,012$ мГн; $C = 1,5$ мкФ; $R_H = 50$ Ом; $f = 200$ кГц; $\gamma = 0,6$. Чому дорівнює вихідна напруга перетворювача при заданих значеннях параметрів моделі? Одержати осцилограму цієї напруги.
2. Скласти модель тиристорного регулятора напруги відповідно до рис. 3.29. Одержати осцилограму напруги однієї фази навантаження при плавній зміні кута керування α від 0 до 120° за час 1 с. Як джерело значення α використовувати блок *Ramp*.
3. Скласти модель однофазного керованого випрямляча (див. рис. 3.21) із системою керування на основі блока *Synchronized 6-pulse generator* (див. рис. 3.31).
4. Скласти модель перетворювача частоти (рис. 3.38), в якому зміна частоти відбувається стрибком з 50 до 20 Гц у момент часу 0,1 с. Реалізувати закон керування $U/f = \text{const}$.

Розділ 4

ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ ПРИ МОДЕЛЮВАННІ ТА ОПТИМІЗАЦІЇ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ

4.1 Завдання теоретичних досліджень

Метою теоретичних досліджень є виділення істотних зв'язків між досліджуваним об'єктом і навколишнім середовищем, пояснення й узагальнення результатів, виявлення загальних закономірностей та їхня формалізація. Теоретичне дослідження завершується формуванням теорії й не обов'язково пов'язане з побудовою теоретичного апарату.

У своєму становленні теорія проходить різні етапи, починаючи від якісного пояснення та кількісного виміру процесів і закінчуючи їхньою формалізацією. Вона може бути представлена як у вигляді якісних правил, так і у вигляді математичних залежностей (керувань, співвідношень).

Завданнями теоретичного дослідження є: узагальнення результатів дослідження, знаходження загальних закономірностей шляхом обробки й інтерпретації дослідних даних; розширення результатів дослідження на ряд подібних об'єктів без повторення всього обсягу досліджень; вивчення об'єктів, недоступних для безпосереднього дослідження; підвищення ймовірності експериментального дослідження об'єкта шляхом обґрунтування параметрів й умов спостереження, точності вимірів.

При проведенні теоретичних досліджень, заснованих на загальнонаукових методах аналізу й синтезу, широко використовуються розчленовування й об'єднання елементів досліджуваної системи (об'єкта, явища, процесу).

Метод розчленовування був запропонований французьким філософом і натуралістом Р. Декартом. У процесі розподілу

виділяються суттєві і несуттєві елементи, а також зв'язки між ними. Р. Декарт звернув увагу на те, що будь-який об'єкт можна розчленувати різними способами й це істотно впливає на проведення теоретичних досліджень. Залежно від обраного виду розчленовування, вивчення може спроститися або, навпаки, ускладнитися, а при неперевіреному підході навіть піти невірним шляхом.

Після розчленовування об'єкта вивчається вид взаємозв'язку елементів і здійснюється їхнє моделювання. У підсумку, моделі елементів поєднуються в складну модель об'єкта.

На всіх етапах побудови моделі об'єкта здійснюється його спрощення та вводяться певні припущення. Припущення обов'язково повинні обґрунтовуватися, тому що невірні припущення можуть призвести до серйозних помилок при формуванні теоретичних висновків.

Протилежним розчленовуванню є метод об'єднання і пов'язаний з ним комплексний підхід до вивчення об'єкта. Такий підхід покладений в основу *загальної теорії систем*, основні положення якої вперше були сформульовані Людвігом фон Берталанфі стосовно вивчення біологічних об'єктів та явищ.

У структурі загальної теорії систем виділяються два напрямки. Перший ставить перед собою завдання розвитку загальної теорії систем як певної філософської концепції, що включає в себе поняття принципу системності, системного підходу, системного аналізу і т. д. У другому напрямку загальна теорія систем являє собою математичний апарат, що претендує на строгий опис закономірностей формування й розвитку будь-яких систем.

Загальна теорія систем ґрунтується на трьох постулатах.

Перший говорить, що функціонування систем будь-якої природи може бути описане на основі розгляду формальних структурно-функціональних зв'язків між окремими елементами системи. Вплив матеріалів, з яких складаються елементи системи, проявляється у формальних характеристиках системи (її структурі, динаміці й тощо).

Другий постулат стверджує, що організація системи може бути визначена на основі проведених ззовні спостережень, за допомогою фіксування станів тільки тих елементів, які безпосередньо взаємодіють з навколишнім середовищем.

Третій постулат полягає у твердженні того, що організація системи повністю визначає її функціонування і характер взаємодії з навколишнім середовищем.

Зазначені постулати дозволяють визначити організацію системи, виходячи з характеру зв'язків між елементами і характеристик взаємодії з навколишнім середовищем.

Прагнення вивчати об'єкти у всіх їхніх зв'язках одержало в загальній теорії систем свій подальший розвиток у формі принципів системності (цілісного подання об'єктів), релятивності систем (будь-яку множину предметів можна розглядати як систему і як не систему) та універсальності. Принцип універсальності спрямований проти абсолютизації окремих систем та способів їхнього утворення стверджує, що будь-яку множину можна розглядати як систему й як не систему лише в певних аспектах та при фіксованих умовах.

Теоретичні дослідження охоплюють аналіз фізичної сутності процесів і явищ, формулювання гіпотези дослідження, побудову й розробку фізичної моделі, проведення математичного дослідження, аналіз теоретичних рішень, формулювання висновків. Якщо немає можливості виконати математичне дослідження, то необхідно сформулювати робочу гіпотезу в словесній формі із залученням графіків, таблиць і т. д.

У технічних науках рекомендується прагнути до використання математичної формалізації висунутих гіпотез і висновків.

Структурно будь-яке науково-дослідне завдання включає в себе умови й вимоги.

Умови – інформація, з якої треба виходити при вирішенні завдання.

Вимоги – мета, яку треба досягти в результаті вирішення умови. Вимоги можуть бути вихідними, притягнутими й шуканими. *Вихідні* вимоги даються в первісному формулюванні завдання. Якщо їх виявляється недостатньо для вирішення завдання, то виникає необхідність у залученні нових даних, що мають назву *притягнуті*.

Шукані дані або шукані умови – це притягнуті умови, які потрібно відшукати в процесі вирішення завдання.

Умови й вимоги завдання перебувають у протиріччі, вони багаторазово зіштовхуються, зіставляються й поступово зближаються між собою. Такі вимоги структурних компонентів завдання тривають доти, поки не буде вирішене саме завдання.

Процес проведення теоретичних досліджень, як правило, складається з декількох стадій.

Оперативна стадія. Ця стадія включає в себе перевірку можливості усунення технічного протиріччя, оцінку можливих змін навколишнього середовища, аналіз можливості переносу розв'язку завдання з інших галузей знання, застосування «зворотного» розв'язку (розв'язок завдання, зворотний даному), аналіз можливості використання природних «праобразів».

Друга, *синтетична стадія*, полягає у визначенні впливу зміни однієї частини об'єкта на побудову інших частин. При цьому визначаються необхідні зміни інших об'єктів, які взаємодіють з даними, оцінюється можливість застосування зміненого об'єкта по-новому, а знайденої технічної ідеї – для вирішення інших завдань.

Успішне проходження цих двох стадій, які можна вважати попередніми, дозволяє приступити до стадії постановки завдання.

При постановці завдання визначається кінцева мета розв'язку, перевіряється можливість досягнення поставленої мети іншими, можливо, більш простими засобами, вибирається найбільш ефективний шлях вирішення завдання і визначаються необхідні кількісні показники.

При необхідності можуть уточнюватися вимоги щодо конкретних умов практичної реалізації передбачуваного розв'язку.

Аналітична стадія включає в себе визначення ідеального кінцевого результату, тобто результату, який би повністю задовольняв кінцеву мету розв'язку. Потім виявляються перешкоди, що заважають отриманню ідеального результату, і знаходяться можливості усунення перешкод.

Чітке формулювання завдання є найважливішим етапом його розв'язання. Перетворення розпливчастого формулювання в чітку, однозначну форму (переформулювання) може істотно полегшити розв'язання.

Розв'язання творчих завдань, безумовно, носить творчий характер. Творчі розв'язки можуть не укладатися в заздалегідь намічені плани. Іноді оригінальні розв'язки з'являються, здавалося б, раптово, після тривалих і безуспішних спроб. Часто вдачі розв'язки пропонуються фахівцями суміжних областей, на яких не мають впливу відомі розв'язки. Це ще раз підтверджує доцільність застосування синергетичного підходу при організації теоретичних досліджень.

Творчі розв'язки, за своєю суттю, – це погляд на явище з нової, незвичної, точки зору.

При розробці теорій важливу роль відіграють логічні методи і правила, що мають нормативний характер. До їхнього числа належать правила висновку, утворення складних понять із простих, установлення істинності складних висловлювань і т. д. Спеціальними способами побудови теорій є принципи формування аксіоматичних теорій, критерії несуперечності, повноти й незалежності систем, аксіом і гіпотез.

Теоретичні дослідження мають велике значення в процесі пізнання об'єктивної дійсності, оскільки забезпечують глибоке розуміння сутності явищ, створюють наукову картину світу, який постійно розвивається. Теоретичне дослідження, будучи функцією мислення, полягає у тому, щоб відкрити, перевірити і засвоїти різні галузі навколишнього світу, створити й розвинути світогляд.

Зростання об'єктивних знань збільшує і сферу питань, що вимагають вирішення. Кожна знайдена відповідь тільки наближає до пізнання істини.

4.2 Математичні методи

Розв'язання завдань теоретичних досліджень математичними методами реалізується шляхом математичного формулювання завдання, вибору методу проведення досліджень отриманої математичної моделі, аналізу отриманого математичного результату.

Математичне формулювання завдання має вигляд чисел, геометричних образів, функцій, систем рівнянь і т. д.

Опис об'єкта або явища може бути представлено безперервною, дискретною, детермінованою, стохастичною або якою-небудь іншою математичною залежністю.

Математична модель – це система математичних залежностей (формул, функцій, рівнянь, систем рівнянь), що описують ту або іншу сторону досліджуваного об'єкта, явища чи процесу.

Першим етапом математичного моделювання є постановка завдання, визначення об'єкта і цілей дослідження, задавання критеріїв (ознак) вивчення об'єкта й керування ним. Неправильна або неповна постановка завдання може нівелювати результати наступних етапів. При цьому важливо встановити границі області впливу досліджуваного об'єкта. Ці границі визначаються областю значимої взаємодії з зовнішніми об'єктами. Така область може бути

визначена на основі наступних ознак: границі області охоплюють ті елементи, вплив яких на досліджуваний об'єкт не дорівнює нулю; за цими границями дія досліджуваного об'єкта на зовнішні об'єкти прагне до нуля.

Врахування області впливу об'єкта дозволяє включити в математичну модель всі істотні фактори і розглядати модельовану систему як замкнуту, тобто з певним ступенем наближення, і незалежну від зовнішнього середовища. У випадку можливості розгляду замкнутої системи математичні дослідження значно спрощуються.

Другий етап полягає у виборі типу математичної моделі. Як правило, послідовно будується кілька моделей. Шляхом порівняння результатів їхнього дослідження з реальними умовами функціонування об'єкта обирається за деяким критерієм найкраща.

При виборі типу математичної моделі з урахуванням даних пошукового експерименту встановлюються: лінійність або нелінійність, тривалість або статичність, стаціонарність або нестаціонарність, а також ступінь детермінованості досліджуваного об'єкта або процесу.

Лінійність встановлюється за характером статичної характеристики досліджуваного об'єкта. Під статичною характеристикою мають на увазі однозначний зв'язок між величиною зовнішнього впливу на об'єкт (вхідний сигнал) і величиною його реакції на зовнішній вплив (амплітуда вихідної характеристики об'єкта або системи).

Під *вихідною характеристикою об'єкта (системи)* розуміють зміни вихідного сигналу в часі. Якщо статична характеристика досліджуваного об'єкта виявляється лінійною, то моделювання цього об'єкта здійснюється з використанням лінійних функцій. Нелінійність статичної характеристики і наявність запізнювання в реакції об'єкта на зовнішній вплив є характерними ознаками нелінійності об'єкта. Використання лінійної математичної моделі значно спрощує її подальший аналіз, оскільки з'являється можливість використання принципу суперпозиції.

Принцип *суперпозиції* стверджує, що при впливі на лінійну систему декількох вхідних сигналів кожний із цих сигналів фільтрується системою так, начебто інші сигнали на неї не діють. Загальний вихідний сигнал лінійної системи, відповідно до принципу суперпозиції, є сумою її реакцій на кожен вхідний сигнал.

Встановлення статичності або динамічності робиться за результатами аналізу поведінки досліджуваних показників об'єкта у часі. Щодо детермінованої системи, висновок про статичність або динамічність може бути зроблений за характером її вихідної характеристики. Якщо середнє арифметичне значення вихідного сигналу за різними відрізками часу не виходить за припустимі межі, які обумовлені точністю методики вимірювання досліджуваного показника, то цей факт може вважатися підтвердженням статичності об'єкта. Стосовно ймовірнісних систем, їхня статичність встановлюється за мінливістю рівня її відносної самоорганізації. У випадку, якщо мінливість цього рівня не перевищує припустимої межі, то система визначається як статична.

Досить важливим є вибір інтервалів часу, на яких встановлюється статичність або динамічність об'єкта. Якщо об'єкт на малих інтервалах часу виявився статичним, то при збільшенні інтервалів результат, як правило, не змінюється. У випадку, якщо статичність встановлена для досить великих часових інтервалів, то при їхньому зменшенні статичність може перейти в динамічність.

При виборі типу моделі ймовірнісного об'єкта важливе значення має встановлення його стаціонарності. Про стаціонарність або нестаціонарність ймовірнісних об'єктів судять за зміною в часі параметрів законів розподілу випадкових величин. Для цього використовують середнє арифметичне випадкової величини $M(\tau_i)$ і середнє квадратичне відхилення випадкових величин σ_i ($i = 1, 2, \dots, n$) від середнього арифметичного відхилення у часі.

З ряду середніх арифметичних $M(\tau_i)$ вибирається мінімальне значення $M(\tau_{\min})$ і будуються інтервали з границями $M(\tau_{\min}) + \Delta x$ і $M(\tau_{\min}) - \Delta x$, де Δx – точність методу вимірювання досліджуваного показника.

Якщо всі значення $M(\tau_i)$ укладаються в цей інтервал, то об'єкт визначається як стаціонарний за середнім арифметичним $M(\tau)$. Аналогічно визначається стаціонарність за середньоквадратичним відхиленням.

Граничні значення σ при встановленні стаціонарності визначаються за формулами

$$\sigma_1 = \sqrt{\frac{D_1}{n}} \quad \text{і} \quad \sigma_2 = \sqrt{\frac{D_2}{n}},$$

де

$$D_1 = \frac{1}{n-1} \sum [x_i - M(\tau_{\min}) + \Delta x]^2;$$

$$D_2 = \frac{1}{n-1} \sum [x_i - M(\tau_{\min}) - \Delta x]^2;$$

n – число спостережень.

Якщо всі значення σ_i укладаються в інтервал значень від σ_1 до σ_2 , то об'єкт вважається стаціонарним. При недотриманні цієї умови об'єкт визначається як ймовірнісний нестаціонарний, навіть якщо величина середнього арифметичного M не змінюється в часі.

Визначення загальних характеристик об'єкта дозволяє вибрати математичний апарат для побудови математичної моделі.

Для детермінованих об'єктів можливе використання апарату лінійної і нелінійної алгебри, диференціального та інтегрального вихування, теорії автоматичного регулювання.

При моделюванні ймовірнісних об'єктів використовуються: теорія детермінованих і випадкових автоматів; теорія випадкових процесів; евристичне програмування; методи теорії інформації; методи теорії керування тощо.

При описі квазидетермінованих (ймовірнісно-детермінованих) об'єктів припустиме використання диференціальних рівнянь з коефіцієнтами, що підкоряються певним законам.

Мета і завдання, поставлені при математичному моделюванні, значною мірою впливають на тип моделі. При вирішенні практичних завдань доцільним є застосування достатньо простого математичного апарату, для фундаментальних – більш складного.

Важливе значення при виборі моделі має аналіз інформаційного масиву, отриманого у результаті аналітичного огляду результатів досліджень інших авторів або пошукового експерименту.

Розподіл масиву на залежні й незалежні фактори, на вхідні і вихідні змінні, попередній пошук взаємозв'язку між різними даними вибірки дозволяє визначити математичний апарат. Аналіз інформаційного масиву дозволяє встановити безперервність або дискретність досліджуваного параметра й об'єкта в цілому.

Так, у безперервних об'єктах всі сигнали є безперервними функціями часу, у дискретних – сигнали квантовані за часом й

амплітудою. Якщо сигнали квантовані тільки за часом, тобто є імпульсами з рівною амплітудою, то об'єкти називають дискретно-безперервними.

Для безперервних об'єктів при моделюванні доцільно використовувати диференціальні рівняння. Дискретність об'єктів вимагає використання апарата теорії дискретних автоматів.

Конкретизація математичної моделі здійснюється з урахуванням мети і завдань, характеру гіпотези й інформаційного масиву, тобто у вихідному типі (класі) моделей визначається її вид.

Вибір виду математичної моделі в даному класі є черговим етапом математичного моделювання. На цьому етапі задається область визначення досліджуваних параметрів, де також необхідно розрахувати значення параметрів, які є припустимими, й установити залежність між ними. Для кількісних параметрів залежності задаються у вигляді систем рівнянь, для якісних – можуть бути використані табличні способи задання функцій.

Якщо параметри описані суперечливими залежностями, то визначаються їх «вагові» коефіцієнти, виражені в балах або частках одиниці. Таким чином, суперечливі залежності стають імовірнісними. При описі складних об'єктів можлива розбивка об'єкта на елементи (підсистеми), установлення ієрархії елементів й опис зв'язків між ними на різних рівнях.

Ключовим моментом при виборі виду математичної моделі є опис перетворення вхідних сигналів у вихідні характеристики об'єкта.

Якщо встановлено, що об'єкт є статичним, то можливе використання алгебраїчних рівнянь, при цьому крім найпростіших алгебраїчних залежностей використовуються регресійні моделі й системи алгебраїчних рівнянь. Якщо характер зміни досліджуваного показника заздалегідь невідомий, то ставиться пошуковий експеримент. Перевага віддається тій математичній формі опису, яка дає найкращий збіг з даними пошукового експерименту. Результати такого експерименту й вихідний інформаційний масив дозволяють установити схему взаємодії об'єкта із зовнішнім середовищем – за співвідношенням вхідних і вихідних величин.

Вибір виду моделі динамічного об'єкта зводиться до складання диференціальних рівнянь. Принципово модель динамічного об'єкта може бути побудована й у класі алгебраїчних функцій, однак за повнотою моделі варто віддавати перевагу моделям, побудованим у класі диференціальних рівнянь.

Якщо досліджувані змінні є тільки функціями часу, то при моделюванні використовуються звичайні диференціальні рівняння. У випадку, якщо змінні залежать і від просторових координат, необхідно використати диференціальні рівняння у частинних похідних. Універсального методу складання диференціальних рівнянь немає, але існують загальні підходи до складання рівнянь першого порядку.

Фізичні й геометричні завдання, як правило, приведуть до одного із трьох видів рівнянь:

- диференціальні рівняння в диференціалах;
- диференціальні рівняння в похідних;
- найпростіші інтегральні рівняння з наступним перетворенням їх у диференціальні рівняння.

Методика складання й розв'язання таких рівнянь докладно описана в спеціальній математичній літературі.

При складанні диференціальних рівнянь регульованих об'єктів необхідно, насамперед, визначити умови одержання рівноважного режиму роботи об'єкта – рівняння статичної рівноваги. Найчастіше рівняння статичної рівноваги виявляється загальним для різних об'єктів дослідження.

Так, при рівномірному русі судна об'єкт буде перебувати в стані статичної рівноваги тільки у випадку, коли рушійні сили F_q дорівнюють силам опору F_c . Рівняння статичної рівноваги має вигляд: $F_q - F_c = 0$.

У випадку, коли об'єкт дослідження обертається (колінчасті вали двигунів внутрішнього згоряння, ротори електродвигунів, генераторів, турбін), умовою статичної рівноваги є рівність крутного моменту M_q моменту опору M_c тоді справедливим буде запис $M_q - M_c = 0$.

Для ємності, в якій необхідно підтримувати постійний рівень рідини, умова статичної рівноваги має вигляд: $V_{пр} - V_{вит} = 0$, де $V_{пр}$ – витрата рідини, яка заливається у ємність, а $V_{вит}$ – витрата виточної з ємності рідини.

Перехідний процес у досліджуваному об'єкті буде спостерігатися при порушенні статичної рівноваги. Збільшення одного з членів рівняння статичної рівноваги викликає збільшення іншого члена, причому ці збільшення, як правило, не рівні між собою. У результаті цього умови статичної рівноваги порушуються й рівняння приймають наступний вигляд:

- для поступального руху $F_q + \Delta F_q \neq F_c + \Delta F_c$;
- для обертового руху $M_q + \Delta M_q \neq M_c + \Delta M_c$;
- для ємності з рідиною $V_{пр} + \Delta V_{пр} \neq V_{виг} + \Delta V_{виг}$ і т. д.

Таким чином, при порушенні умов статичної рівноваги в досліджуваному об'єкті виникає надлишок або нестача рушійних сил, моментів, надходження рідини і т. п. Подальше перетворення отриманих нерівностей здійснюється шляхом залучення відомих залежностей, принципів, законів або аналізу вихідних характеристик об'єктів, отриманих при проведенні пошукового експерименту. При цьому можуть бути використані феноменологічні закони (установлені й загальновизнані раніше), напівемпіричні та чисто емпіричні співвідношення.

Так, при поступальному русі об'єкта порушення статичної рівноваги приведе до прискореного або уповільненого руху. Оскільки будь-який об'єкт, що рухається, має масу m із прискоренням a , то на підставі принципу Даламбера рівняння динамічної рівноваги прийме вигляд:

$$ma = \Delta F_q - \Delta F_c.$$

Оскільки прискорення $a = \frac{dv}{d\tau}$ (v – лінійна швидкість об'єкта; τ – час), тоді можна записати рівняння для поступального руху об'єкта: $m \frac{dv}{d\tau} = \Delta F_q - \Delta F_c$.

Для об'єкта, що має обертовий рух, при одержанні рівняння динамічної рівноваги так само використовується принцип Даламбера. Тут рівняння руху записується так:

$$J \frac{d\omega}{d\tau} = \Delta M_q - \Delta M_c,$$

де J – приведений до осі обертання ведучого вала момент інерції деталей механізму; ω – кутова швидкість обертання ведучого вала.

Аналогічним чином можуть бути отримані рівняння динамічної рівноваги і для інших об'єктів.

Порівняння диференціальних рівнянь для різних регульованих об'єктів свідчить про ідентичність їхньої структури.

Опис збільшень (ΔF , ΔM) у розглянутих рівняннях вимагає інформації про їхній зв'язок із властивостями об'єкта. Внаслідок

цього, введення в рівняння динамічної рівноваги залежностей, що описують розглянуті збільшення, може привести до підвищення порядку диференціальних рівнянь. З метою спрощення, шляхом зниження порядку диференціальних рівнянь, найчастіше використовують лінеаризацію збільшень.

При лінеаризації, як правило, використовується розкладання функції у ряд Маклорена. Так, наприклад, обертовий момент вала дизеля залежить від кутової швидкості обертання ω і положення органа керування паливним насосом h . Тоді можна записати $M_q = f(\omega, h)$. При визначенні збільшення крутного моменту двигуна, залежно від збільшення ω і h , отриману функцію розкладемо у ряд Маклорена:

$$M_q + \Delta M_q = M_q + \frac{\partial M_q}{\partial \omega} d\omega + \frac{\partial^2 M_q}{\partial \omega^2} \times \\ \times \frac{\partial \omega^2}{2!} + \dots + \frac{\partial M_q}{\partial h} dh + \frac{\partial^2 M_q}{\partial h^2} \cdot \frac{\partial h^2}{2!} + \dots$$

При заміні нескінченно малих величин $d\omega$ і dh досить малими скінченними величинами $\Delta\omega$ і Δh ряд прийме вигляд:

$$M_q + \Delta M_q = M_q + \frac{\partial M_q}{\partial \omega} \Delta\omega + \frac{\partial^2 M_q}{\partial \omega^2} \times \\ \times \frac{\Delta\omega^2}{2!} + \dots + \frac{\partial M_q}{\partial h} \Delta h + \frac{\partial^2 M_q}{\partial h^2} \cdot \frac{\Delta h^2}{2!} + \dots$$

Похідні в цьому розкладанні підраховуються для точки, де виконується умова статичної рівноваги.

При малих скінченних збільшеннях $\Delta\omega$ і Δh та нерозривності функції $M_q = f(\omega, h)$ можна відкинути всі члени ряду, які містять $\Delta\omega$ і Δh у степенях вище першої, що не внесе істотної помилки. При виконанні таких перетворень дійсна функція $M_q = f(\omega, h)$ практично замінюється її дотичною у точці рівноважного режиму, що й називається лінеаризацією.

У результаті лінеаризації збільшення крутного моменту можна представити в наступному вигляді: $\Delta M_q = \frac{\partial M_q}{\partial \omega} \Delta\omega + \frac{\partial M_q}{\partial h} \Delta h$. Якщо момент опору є функцією тільки кутової швидкості $M_c = f(\omega)$, то

після розкладання цієї функції в ряд Маклорена й наступної лінеаризації одержимо: $\Delta M_c = \frac{dM_c}{d\omega} \Delta\omega$.

Рівняння динамічної рівноваги об'єкта після підстановки приймає вигляд:

$$J \frac{d\omega}{dt} + \left(\frac{\partial M_c}{\partial \omega} - \frac{\partial M_g}{\partial \omega} \right) \Delta\omega = \frac{dM_g}{dh} \Delta h.$$

Будь-які диференціальні рівняння є моделлю класу явищ, характеризованих аналогічними процесами. У результаті інтегрування одержують велику кількість розв'язків, що задовольняють вихідному диференціальному рівнянню. Для одержання розв'язку, що задовольняє розглянутому процесу, необхідно задати додаткові умови. Ці умови називаються *умовами однозначності* й повинні розкривати особливості даного рівняння стосовно досліджуваного об'єкта.

Умови однозначності характеризуються наступними ознаками: геометрією системи (форма й розміри); фізичними властивостями (теплопровідність, пружність, вологопровідність і т. д.); початковими умовами, тобто станом системи у момент початку дослідження; граничними умовами, що визначають взаємодію системи на границях з навколишнім середовищем. Початкові й граничні умови називаються *крайовими*.

Розберемо приклади вибору, які застосовуються до об'єктів, що можуть розглядатися як детерміновані. При виборі виду математичної моделі для імовірнісних об'єктів, як і у випадку детермінованих, під стаціонарністю входу розуміють його постійне значення. У випадку, якщо сигнал приймає кілька значень, то його вважають не-стаціонарним.

У найбільш простому випадку, якщо вплив на вхід об'єкта постійний у часі, то як математична модель статичного імовірнісного об'єкта може бути прийнятий деякий закон розподілу вихідної величини. Якщо вхідний вплив приймає різні значення, кожному з яких відповідає ряд значень вихідної величини об'єкта, як модель приймається набір законів розподілу вихідної величини для всіх значень вхідного впливу.

У ході моделювання імовірнісних об'єктів поряд із законами розподілу вхідних і вихідних величин необхідно враховувати зв'язок між ними. Із цією метою до складу моделі включають

коефіцієнти взаємної кореляції та функції: $H_m = f(x)$; $R = f(x)$; $y_{\text{ср}} = f(x)$; $\sigma = f(x)$, де x – вхідний вплив; H_m – максимальна ентропія вихідних характеристик; R – відносна організація вихідних характеристик; $y_{\text{ср}}$ – середнє значення вихідної величини; σ – середньоквадратичне відхилення вихідних величин.

Максимальна ентропія вихідних характеристик оцінюється залежністю $H_m = \log_2 n$, де n – число можливих станів об'єкта. Число станів об'єкта визначається як $n = \frac{y_{\text{max}} - y_{\text{min}}}{\Delta y}$, де y_{max} , y_{min} – максимальне й мінімальне значення вихідних величин; Δy – точність виміру цих величин.

Відносна організація вихідних характеристик оцінюється за формулою Ферстера: $R = 1 - \frac{H}{H_m}$, де $H = - \sum_{i=1}^N \frac{m_i}{N} \log_2 \frac{m_i}{N}$; m_i – число появ значення вихідної характеристики y_i ; N – повне число спостережень вихідних характеристик.

Нестационарність відрізняється тим, що кожний вплив може приймати кілька значень. У цьому випадку, для кожного конкретного сполучення, завдання аналізу зв'язку може вирішуватися аналогічно завданню для стаціонарної схеми.

Оцінка ступеня зв'язку виходів із входами проводиться шляхом зіставлення параметрів й обчислення коефіцієнтів взаємної кореляції.

При моделюванні динамічних режимів імовірнісних об'єктів на вхід багаторазово надходить сигнал однієї і тієї ж функції часу $x(\tau)$, а на виході спостерігається стаціонарний або нестационарний випадковий процес.

У випадку стаціонарного вихідного процесу математичною моделлю вихідної величини приймається закон розподілу значень вихідної величини, що має ті самі параметри для всіх зрізів за часом. Зрізи за часом приймаються з інтервалом $\Delta\tau$, а модель доповнюється залежностями:

$$H_m(i, \Delta\tau) = f(x); \quad R(i, \Delta\tau) = f(x), \quad i = 1, 2, 3, \dots$$

Ці залежності можуть бути представлені як алгебраїчною функцією, так і диференціальними рівняннями.

У випадку нестационарного виходу об'єкта, математична модель подається у вигляді функціональних зв'язків:

$$m_y(i, \Delta t) = f(x) \quad \text{та} \quad \sigma_y(i, \Delta t) = f(x), \quad i = 1, 2, 3 \dots,$$

де $m_y(i, \Delta t)$ – математичне очікування розподілу вихідних величин з початком Δt , а $\sigma_y(i, \Delta t)$ – середньоквадратичне відхилення.

Для опису таких залежностей також можуть використовуватися алгебраїчні або диференціальні рівняння.

При впливі на об'єкт декількох факторів й оцінки його поведінки за рядом показників у нестационарному режимі, необхідно встановити функціональні залежності імовірнісних характеристик для кожного входу й виходу. При цьому необхідно враховувати, що будь-який вихід залежить від всіх входів.

Процес вибору математичної моделі завершується її попереднім контролем. Необхідно провести наступні види контролю: розмірності; порядків; характеру залежностей; екстремальних ситуацій; граничних умов; математичної замкнутості; фізичного змісту; стійкості.

Контроль розмірності полягає у виконанні правила, відповідно до якого прирівнюватися й алгебраїчно додаватися можуть тільки величини однакової розмірності.

Контроль порядків полягає у визначенні порядку величин, які додаються, при цьому малозначимі доданки відкидаються, що дозволяє спростити модель.

Контроль характеру залежностей зводиться до перевірки напрямку й швидкості зміни одних величин при зміні інших. Напрямки й швидкості, враховані математичною моделлю, повинні відповідати фізичному змісту завдання.

Контроль екстремальних ситуацій полягає в перевірці наочно-го змісту розв'язку при прагненні параметрів моделі до нуля або нескінченності.

Контроль граничних умов має на увазі перевірку відповідності математичної моделі граничним умовам, що впливають зі змісту завдання. Перевіряються, чи дійсно граничні умови поставлені, чи враховані при побудові шуканої функції, чи задовольняє функція таким умовам.

Контроль математичної замкнутості зводиться до перевірки наявності однозначного розв'язку математичної моделі.

Контроль фізичного змісту проводиться для перевірки фізичного змісту проміжних співвідношень, що використовуються при побудові моделі.

Контроль стійкості моделі полягає в перевірці впливу варіювання наявних даних про реальний об'єкт на зміну результатів математичного моделювання.

4.3 Аналітичні методи

Під аналітичними методами розуміють сукупність методів, які використовуються для дослідження обраної математичної моделі. При цьому керуються принципом відповідності зовнішньої та внутрішньої правдоподібності.

Під *зовнішньою правдоподібністю* розуміється очікуваний ступінь адекватності математичної моделі реальному об'єкту за досліджуваними властивостями.

Під *внутрішньою правдоподібністю* розуміється очікуваний ступінь точності розв'язків рівнянь, які прийняті як математична модель об'єкта.

Суть принципу відповідності зовнішньої і внутрішньої правдоподібності полягає у тому, що ступінь точності обчислень повинен відповідати ступеню точності вихідних даних. Якщо вид моделі попередньо обраний, то зовнішню правдоподібність вважають фіксованою і вибір методу дослідження визначається необхідним ступенем внутрішньої правдоподібності.

У деяких випадках можливе відхилення від зазначеного принципу. Це може бути доцільно у таких випадках:

- при розробці нового єдиного методу досліджень для широкого класу моделей, при цьому необхідно прагнути до максимальної внутрішньої правдоподібності дослідження незалежно від рівня зовнішньої правдоподібності;

- при перевірці зовнішньої правдоподібності моделі внутрішня правдоподібність методу перевірки повинна бути максимальною;

- у випадку досить простої моделі й можливості точного розв'язку немає потреби в штучному зниженні строгості розв'язку.

Ефективність вибору методу розв'язку багато в чому залежить від наявності передбачуваного кінцевого розв'язку завдання. Такі відомості можуть бути отримані при пробних дослідженнях моделі та її елементів. У процесі проб здійснюється порівняння величин окремих членів рівнянь у досліджуваному діапазоні зміни параметрів. Відносно малі параметри відкидаються, нелінійні залежності

заміняються лінійними, деякі з компонентів моделі апроксимуються наближеними рівняннями. Такі дії дозволяють швидко одержати приблизний розв'язок завдання. Приблизне знання якісних і кількісних характеристик шуканого розв'язку допомагає при виборі точності методу дослідження.

Вибір методу дослідження математичної моделі в значній мірі визначається її видом.

Статичні системи, представлені алгебраїчними рівняннями, досліджуються за допомогою визначників, методу ітерацій, методів Крамера, Гаусса, можливе застосування наближених графічних методів.

При дослідженні динамічних систем, описуваних диференціальними рівняннями, широко використовуються метод поділу змінних, метод підстановки, метод інтегруючого множника, метод якісного аналізу та інших. Для одержання наближених розв'язків використовують метод послідовних наближень, метод функціональних рядів і числові методи інтегрування. Для докладного вивчення моделей динамічних систем, побудованих у класі диференціальних рівнянь, використовується теорія диференціальних рівнянь.

Використання аналітичних методів розв'язання математичних завдань є основним методом наукового дослідження. При цьому складність прийнятих методів розв'язання рівнянь ускладнює отримання кінцевих розв'язків. У зв'язку з цим при розв'язанні практичних завдань широко застосовуються методи перетворення вихідних рівнянь (логарифмування, Лапласа, перетворення Фур'є і т. д.).

Найпростішим способом перетворень є логарифмування рівнянь.

Нехай необхідно одержати розв'язок простого рівняння $y = a^{0.4}$. Таке рівняння називається оригіналом функції. Зведення числа a в степінь $0,4$ прямими методами складне. Логарифмуючи, одержимо $\lg y = 0,4 \lg a$. Це рівняння називається зображенням функції. При логарифмуванні функція переводиться із простору оригіналів у простір зображень, що дозволяє операцію зведення в степінь замінити множенням показника степеня на $\lg a$. За допомогою антилогарифмування отриманий результат переводиться із простору зображень у простір оригіналів. Зміст перетворень Лапласа і Фур'є аналогічний логарифмуванню.

У перетвореннях Лапласа вихідна функція часу переводиться із простору оригіналів у простір зображень за допомогою інтеграла

$$F(p) = \int f(\tau) \cdot e^{-p\tau} \cdot d\tau,$$

де p – оператор Лапласа (можлива формальна заміна $p \Leftrightarrow d/d\tau$).

Переведення функції із простору зображень у простір оригіналів здійснюється за допомогою інтеграла $F(\tau) = \frac{1}{2\pi} \int_{c-i\infty}^{c+i\infty} F(p) \cdot e^{p\tau} \cdot dp$.

Перетворення Лапласа широко використовується при розв'язанні диференціальних й інтегральних рівнянь, у теорії автоматичного керування. Ґрунтуючись на методі перетворення функцій, вирішуються завдання аналізу перехідних процесів у системах керування. У процесі аналізу використовуються передатні функції.

Під передатною функцією розуміються відношення перетворення Лапласа вихідної координати лінійної системи до перетворення Лапласа вхідної координати при нульових початкових умовах: $W(p) = \frac{y(p)}{x(p)}$, де $W(p)$ – передатна функція; $x(p)$ і $y(p)$ – перетворення Лапласа вхідного й вихідного зображень сигналів відповідно.

Передатна функція може бути отримана з диференціального рівняння шляхом заміни операції диференціювання за часом оператором Лапласа p , а операції інтегрування за часом – оператором p^{-1} .

Нехай система описується диференціальним рівнянням:

$$\frac{dy(\tau)}{d\tau} + a \cdot y(\tau) = k \cdot x(\tau).$$

Переходячи до зображень, одержуємо:

$$p \cdot y(p) + a \cdot y(p) = k \cdot x(p).$$

Таке подання диференціального рівняння називається операційним. Для знаходження передатної функції цієї системи досить зробити алгебраїчні перетворення: $W(p) = \frac{y(p)}{x(p)} = \frac{k}{p+a}$.

Поряд з методом передатних функцій для аналізу систем керування використовується метод частотних характеристик.

Частотні характеристики систем застосовують при аналізі стійкості, якості перехідних процесів, динамічної точності, синтезі коригувальних пристроїв. Сутність методу частотних характеристик полягає в урахуванні того, що кожен елемент, динамічні властивості якого характеризуються лінійним диференціальним рівнянням, при гармонічних коливаннях ω вхідної координати виробляє на виході гармонічні коливання з тією же частотою ω , але з іншою амплітудою і фазовим зсувом. Іншими словами, якщо графіки гармонічних коливань сполучити, то виявиться, що коливання вихідної координати φ відбуваються з тією ж частотою ω , своєю амплітудою φ_0 і деяким зсувом фази $\gamma(\omega)$, причому амплітуда φ_0 і зсув фази γ залежать від частоти ω збудовального впливу. Побудована характеристика кожною своєю точкою показує амплітуду коливань і зсув фази в порівнянні з коливаннями вхідної координати.

У кожній точці такої характеристики, отриманої експериментальним або розрахунковим шляхом, знаходять значення частоти коливань, амплітуди й фази. Амплітудно-фазова частотна характеристика (АФЧХ) здатна охопити всі можливі гармонічні коливання, тому що може будуватися для будь-яких частот коливань вхідної координати.

При необхідності дослідження випадкових, ймовірнісних (стохастичних) процесів доводиться аналізувати випадкові стохастичні зв'язки, у яких кожному аргументу відповідає безліч значень функції. У таких випадках теорія ймовірності дозволяє представити результат не однієї якої-небудь події, а середній результат випадкових подій. І тим точніше, чим більше число аналізованих явищ: незважаючи на випадковий характер подій, вони підкоряються певним законам *теорії ймовірностей*.

Теорія ймовірностей вивчає випадкові події й базується на наступних основних показниках:

- сукупність безлічі однорідних подій випадкової величини x є первинним статистичним матеріалом;
- сукупність, що містить різні варіанти масового явища, показують генеральною сукупністю або великою вибіркою N ;
- як правило, вивчається частина генеральної сукупності, яка називається вибірковою сукупністю або малою вибіркою N_1 ;
- ймовірністю $p(x)$ події x називають відношення числа випадків $N(x)$, яке приводить до настання події x , до загального числа можливих випадків N , тобто $p(x) = N(x) / N$.

Таким чином, теорія ймовірності розглядає теоретичний розподіл випадкових величин, їхні характеристики і разом з математичною статистикою становить єдину математичну теорію масових випадкових процесів, що застосовується в наукових дослідженнях.

У математичній статистиці ключовим є поняття про частоту подій $\bar{y}(x)$, що є відношенням числа випадків $n(x)$, при яких мала місце подія, до загального числа подій n : $\bar{y}(x) = n(x) / n$. При необмеженому зростанні числа подій $y(x)$ прагне до ймовірності $p(x)$.

Частота $y_{io} = n(x) / \sum n(x)$ характеризує ймовірність появи випадкової величини і являє собою ряд розподілу (гістограму), а плавна апроксимуюча крива – закон або функцію розподілу $F(x)$. Ймовірність випадкової величини (події) – кількісна оцінка можливості її появи. Достовірна подія має ймовірність $p = 1$, неможлива подія – ймовірність $p = 0$, випадкова подія – ймовірність $0 \leq p(x) \leq 1$, а сума ймовірностей всіх можливих значень – $\sum_{i=0}^n p_i = 1$.

Крім функції розподілу, необхідно мати такі характеристики, як середньоарифметичне й математичне очікування, дисперсія й розмах ряду розподілу.

Якщо серед n подій величина x_1 повторюється n_1 разів, величина x_2 повторюється n_2 разів і так далі, середнє арифметичне значення x визначається як $x = \sum_{i=1}^n x_i / n$. Розмах використовується для орієнтовної оцінки ряду подій:

$$R = x_{\max} - x_{\min},$$

де x_{\max} та x_{\min} – максимальне й мінімальне значення вимірюваної величини або похибки.

Якщо замість емпіричних частот y_p, \dots, y_n прийняти їхні відповідні ймовірності p_p, \dots, p_n , то одержимо таку характеристику, як *математичне очікування*: $m(x) = \sum_{i=1}^n x_i \cdot p_i$.

Для безперервних випадкових величин математичне очікування визначається інтегралом $m(x) = \int x \cdot p(x) \cdot dx$, тобто воно дорівнює дійсному значенню x_q спостережуваних подій. Таким чином, якщо

систематичні похибки вимірювань повністю виключені, тоді істинне значення вимірюваної величини дорівнює математичному очікуванню, а відповідна йому абсциса на графіку функції розподілу називається центром розподілу. Мірою розсіювання (точності вимірів) є дисперсія $D(x)$. Дисперсія обчислюється за формулою

$$D(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$
, де x – аналізований показник; \bar{x} – його середнє значення; n – кількість значень показника в аналізованій сукупності даних.

Середньоквадратичне відхилення $\sigma(x)$ характеризує розсіювання випадкової величини стосовно математичного очікування. Середньоквадратичне відхилення називають також стандартним відхиленням і розраховують за виразом $\sigma(x) = \sqrt{D(x)}$.

При використанні ймовірнісних методів важливим є вибір закону розподілу. При відсутності достатнього обсягу даних вибір повинен ґрунтуватися або на минулому досвіді, або на знанні конкретного фізичного механізму виникнення випадку, що відповідає конкретній події x . Якщо, припустимо, прийняти пропозицію про усереднення ймовірності виникнення події x на всьому інтервалі вивчення досить простого об'єкта, наприклад елемента системи, можна рекомендувати нормативний розподіл. Якщо вплив збурювання на систему з часом змінюється, наприклад у зв'язку зі «старінням» системи, а ймовірність події x при цьому збільшується, використовується експонентний розподіл.

При врахуванні зміни в часі гранично допустимих значень контрольованої величини x_{\max} й x_{\min} , перевищення яких безумовно приведе до події, наприклад до відмови досліджуваного елемента або системи, кращим виявляється розподіл Вейбулла, так само можуть бути використані гамма-розподіл або розподіл Релея. Більш докладно вибір закону розподілу випадкової величини описаний у відповідній літературі [12].

У дослідженнях часто виникає необхідність виявлення факторів та їхніх комбінацій, які суттєво впливають на досліджуваний процес, тому що при вимірі певної величини результати можуть залежати від ряду факторів. Для технічних об'єктів і систем основними факторами є технічний стан приладу й увага оператора. Для встановлення основних факторів та їхнього впливу на досліджуваний

процес використовується дисперсійний одно- та багатofакторний аналіз.

Методи теорії ймовірностей і математичної статистики часто застосовуються в теорії надійності.

Під *надійністю* розуміється властивість об'єкта виконувати задані функції протягом необхідного періоду часу. В теорії надійності відмови розглядаються як випадкові події. Для кількісного опису відмов застосовуються математичні моделі – функції розподілу ймовірностей інтервалів часу. Як правило, використовуються закони нормального й експонентного розподілу або закон Вейбулла. Основним завданням теорії надійності є прогнозування різних показників безвідмовної роботи, що пов'язано із знаходженням ймовірностей.

При дослідженнях систем з метою відшукування найбільшої ефективності задоволення умови «вимога–обслуговування» застосовуються методи теорії масового обслуговування. Основними характеристиками теорії масового обслуговування є: інтенсивність надходження вимог на обслуговування λ , інтенсивність обслуговування μ , коефіцієнт використання системи $\phi = \lambda/\mu$, час очікування обслуговування t_o , тривалість обслуговування t_m , час обслуговування в системі t_{ob} , число вимог у черзі n , математичне очікування числа вимог у системі n_c .

При розрахунку приймаються середні випадкові значення λ , μ , t_o , t_m , t_{ob} , n . Розрахунок часу обслуговування за тривалістю звичайно виражають показниковим законом. Завданням теорії масового обслуговування є встановлення достовірних залежностей між інтенсивністю потоку вимог і продуктивністю (пропускною здатністю) системи.

При *оптимізації* різних процесів можливе використання методів теорії ігор, яка розглядає розвиток процесів залежно від випадкових ситуацій. Теорія ігор дозволяє оцінювати найбільш сприятливі та несприятливі ситуації і на їх основі приймати оптимальне для даних умов розв'язання.

Розв'язання задачі оптимізації аналітичними методами полягає у знаходженні екстремального значення відомої функції $f(x_1, x_2, \dots, x_n)$ в області значень її параметрів x_1, x_2, \dots, x_n . Необхідно враховувати, що при розв'язанні оптимізаційних завдань можливі випадки, коли наслідком оптимізації одного процесу є погіршення

іншого. Тому при оптимізації варто дотримуватися комплексності розв'язання, при якій досліджуються всі особливості процесу (див. у відповідній літературі критерії Парето).

Сучасні програмні продукти, орієнтовані на автоматизоване проектування, дозволяють використовувати різні методи оптимізації для систем і пристроїв, що описуються у рамках даного програмного середовища не тільки аналітично, але й у вигляді структурних схем, у тому числі з нелінійними елементами.

Розглядаючи завдання по етапах, необхідно оцінювати ситуацію в цілому, яка може змінюватися у результаті оптимізації досліджуваного процесу.

Цільова функція в оптимізаційних (екстремальних) задачах – функція невідомих, екстремум (максимум чи мінімум) якої необхідно знайти. Якщо він існує, то сукупність тих невідомих задач, за яких цільова функція досягає шуканого екстремуму, називається розв'язком (інколи розв'язком називають саме значення екстремуму). Вибір показника, формалізований опис якого буде прийнято за цільову функцію, – найважливіша частина змістовності роботи у постановці оптимізаційної задачі. Формально цільова функція задається по-різному – залежно від обраних математичних засобів опису задачі. Як синоніми цільової функції застосовуються:

- а) функція мети;
- б) показник якості розв'язку;
- в) критерій оптимізації;
- г) критерій якості тощо.

Аналітичні методи оптимізації застосовуються для розв'язання різних наукових завдань у тому випадку, коли функція мети й обмеження задані аналітично – у вигляді безперервних диференційованих функцій.

Розглянемо деякі варіанти розв'язання завдань оптимізації.

4.4 Методи оптимізації

4.4.1 Аналітичні методи оптимізації

Знаходження екстремуму функції однієї змінної без обмежень. Нехай залежність критерію від змінної x задана безперервною диференційованою функцією

$$F = F(x). \quad (4.1)$$

Як відомо з математичного аналізу, необхідна умова екстремуму функції $F(x)$ – рівність нулю її першої похідної:

$$\frac{dF}{dx} = 0. \quad (4.2)$$

Точка x^0 , у якій похідна перетворюється в нуль (дотична до функції $F(x)$, паралельна осі абсцис), називається стаціонарною точкою функції $F(x)$. Стаціонарна точка може бути максимумом, мінімумом або точкою перегину (рис. 4.1, а). Тип стаціонарної точки визначається достатніми умовами оптимальності.

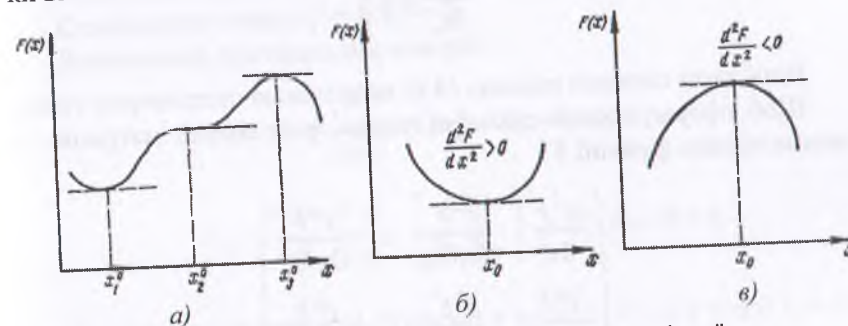


Рисунок 4.1 – Екстремуми функції однієї змінної

Якщо друга похідна $F(x)$ у точці x^0 додатна, то в цій точці є мінімум (рис. 4.1, б):

$$\frac{d^2F}{dx^2} > 0. \quad (4.3)$$

Якщо друга похідна від'ємна – у точці x^0 є максимум (рис. 4.1, в):

$$\frac{d^2F}{dx^2} < 0. \quad (4.4)$$

Приклад: функція задана аналітичним виразом $F(x) = 3x^2 - 2x - 1$. Маємо $\frac{dF}{dx} = 6x - 2 = 0$; $x_0 = \frac{1}{3}$; $\frac{d^2F}{dx^2} = 6 > 0$. У точці $x_0 = \frac{1}{3}$ – мінімум.

Екстремум функції багатьох змінних. Нехай залежність критерію від змінних задається безперервною диференційованою функцією

$$F = F(x_1, x_2, \dots, x_n). \quad (4.5)$$

Необхідною умовою екстремуму є рівність нулю частинних похідних функції F по всіх змінних, тобто

$$\left. \begin{aligned} \frac{\partial F}{\partial x_1} &= 0; \\ \frac{\partial F}{\partial x_2} &= 0; \\ &\dots \\ \frac{\partial F}{\partial x_n} &= 0. \end{aligned} \right\} \quad (4.6)$$

Розв'язком системи рівнянь (4.6) визначаємо стаціонарну точку. Щоб сформулювати достатні умови, розглянемо матрицю інших похідних функції F :

$$M = \begin{pmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_1 \partial x_2} & \frac{\partial^2 F}{\partial x_2^2} & \dots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_n^2} \end{pmatrix}. \quad (4.7)$$

Запишемо визначники діагональних мінорів цієї матриці як $\Delta_1, \Delta_2, \dots, \Delta_n$:

$$\Delta_1 = \frac{\partial^2 F}{\partial x_1^2}; \quad \Delta_2 = \begin{vmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} \\ \frac{\partial^2 F}{\partial x_1 \partial x_2} & \frac{\partial^2 F}{\partial x_2^2} \end{vmatrix} = \frac{\partial^2 F}{\partial x_1^2} \cdot \frac{\partial^2 F}{\partial x_2^2} - \left(\frac{\partial^2 F}{\partial x_1 \partial x_2} \right)^2 \quad (4.8)$$

і так далі.

Щоб у стаціонарній точці був мінімум, достатньо, щоб визначники всіх діагональних мінорів були більше нуля:

$$\Delta_1 > 0; \Delta_2 > 0; \Delta_3 > 0, \dots, \Delta_n > 0. \quad (4.9)$$

Для того щоб у стаціонарній точці був максимум, достатньо, щоб визначники всіх парних мінорів були більше нуля, а всіх непарних мінорів – менше нуля:

$$\Delta_1 < 0; \Delta_2 > 0; \dots, \Delta_{2k} > 0, \dots, \Delta_{2k+1} < 0. \quad (4.10)$$

Приклад: задана функція двох змінних $F = x_1^4 + 2x_2^2 - 4x_1$. Тоді визначимо

$$\frac{\partial F}{\partial x_1} = 4x_1^3 - 4 = 0, \quad \frac{\partial F}{\partial x_2} = 4x_2 = 0.$$

Стаціонарна точка $x_1^0 = 1; x_2^0 = 0$.

Визначники діагональних мінорів:

$$\Delta_1 = \frac{\partial^2 F}{\partial x_1^2} = 12x_1^2 = 12 > 0;$$

$$\Delta_2 = \frac{\partial^2 F}{\partial x_1^2} \cdot \frac{\partial^2 F}{\partial x_2^2} - \left(\frac{\partial^2 F}{\partial x_1 \partial x_2} \right)^2 = 12 \cdot 4 = 48 > 0.$$

Відповідно до умови (4.9) маємо в точці $x_1 = 1$, а в точці $x_2 = 0$ – мінімум.

Існує досить багато різних аналітичних методів пошуку екстремумів функцій, заданих аналітично (з обмеженнями у вигляді рівностей і нерівностей, на основі множників Лагранжа та ін.). Ознайомитися з цими методами можна в численній літературі.

Недоліки аналітичних методів оптимізації. Аналітичні методи оптимізації, засновані на необхідних умовах екстремуму, не завжди застосовуються з наступних причин.

1. Для розв'язання завдань аналітичними методами функція мети і модель об'єкта повинні описуватися безперервними гладкими диференційованими функціями, заданими в аналітичній формі. Реальні моделі об'єктів часто описуються негладкими недиференційованими функціями, наприклад кусково-лінійними, або бувають задані у вигляді таблиць і т. д. У цих випадках, перед розв'язанням, необхідно попередньо апроксимувати ці залежності гладкими функціями.

2. При використанні аналітичних методів максимальне значення функції визначається в її стаціонарній точці. Проте стаціонарних

точок може бути декілька. Тоді доводиться робити перевірку у всіх стаціонарних точках, відшукуючи найбільше значення (глобальний максимум) функції. Максимум взагалі може й не бути, наприклад у випадку лінійної функції (рис. 4.2, а). Крім того, найбільше значення функції мети взагалі може досягатися не в стаціонарній точці, а на межі області визначення функції мети (рис. 4.2, б).

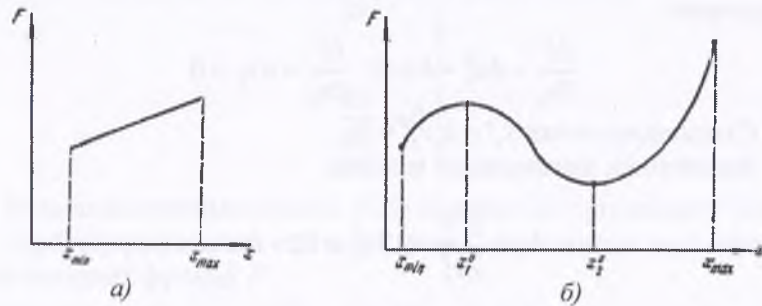


Рисунок 4.2 – Лінійна функція мети (а), максимум знаходиться на межі області визначення і в стороні від локальних максимумів (б)

3. Велика розмірність, наявність обмежень у вигляді рівностей й особливо у вигляді нерівностей суттєво ускладнюють розв'язання завдання оптимізації.

4. Аналітичні методи часто погано пристосовані для розв'язання задач оптимізації за допомогою ЕОМ.

4.4.2 Чисельні методи оптимізації

Існують методи оптимізації, які засновані на поступовому переміщенні в просторі змінних у напрямку до найбільшого значення функції мети. Головна особливість таких методів полягає у тому, що найбільше значення функції шукається (визначається) не відразу, а в результаті процесу переміщення, який розгортається у часі як наслідок певної послідовності дій. Різні методи пошуку вирізняються способом визначення шляху до оптимуму.

Розглянемо деякі варіанти пошуку на прикладі функції двох змінних. Отримані результати можна буде узагальнити на випадок багатьох змінних.

На рис. 4.3 зображена площина варіації змінних x_1 і x_2 . Кривими лініями на цій площині зображені поверхні рівня функції $F(x_1, x_2)$,

тобто поверхні, на яких функція мети F має постійне значення. Точка x^0 з координатами x_1^0 і x_2^0 є максимумом функції мети F . Пошук точки x^0 здійснюється різними методами.

Метод сканування. Метод сканування (або метод повного перебору) полягає в тому, що обчислюють всі значення функції F_i у вузлах квадратної сітки з кроком h (на рис. 4.4 вузли сітки зображені жирними точками). З обчислених значень F_i вибирають найбільше F_0 . Координати відповідного вузла сітки x_1^0 і x_2^0 – координати екстремуму, визначені з точністю до $\pm h$. Чим менше крок сітки, тим вище точність обчислення, однак при зменшенні кроку швидко зростає необхідна кількість обчислень. Дійсно, якщо інтервал зміни кожної змінної розбитий на 10 рівних частин, то при двох змінних необхідна кількість обчислень функції F дорівнює 100, при трьох змінних – 1000, при n змінних – 10^n .

Тому метод повного перебору застосовується тільки при малому числі змінних (у випадку одновимірного або двовимірного об'єкта).

Релаксаційний метод (метод Гаусса). З початкової точки A_0 з координатами x_{10} і x_{20} переміщуємося, змінюючи значення однієї змінної, а значення іншої змінної при цьому залишаємо постійним. На рис. 4.5 змінюється значення змінної x_1 , а змінна x_2 залишається постійною й рівною x_{20} . Переміщення триває доти, поки функція F зростає до точки $A_1(x_{10}, x_{20})$. У цій точці напрямок руху змінюється. Тепер перша змінна залишається постійною і рівною x_{11} , друга змінна змінюється в напрямку зростання функції F до точки $A_2(x_{11}, x_{21})$. У цій точці знову змінюється напрямок

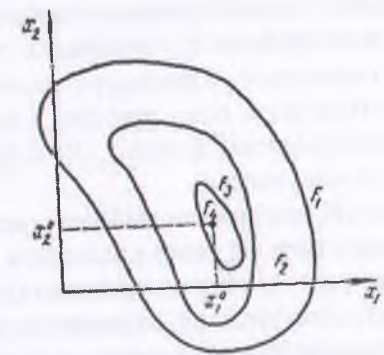


Рисунок 4.3 – Лінії рівня функції мети

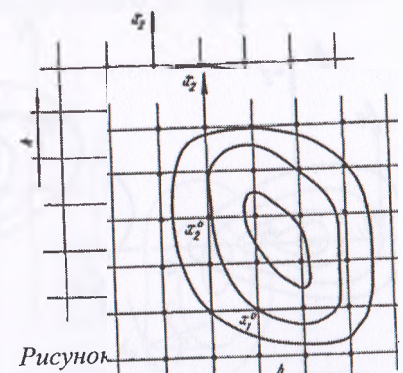


Рисунок 4.4 – Метод повного перебору

руху. Змінна x_2 залишається постійною і рівною x_{21} , а змінна x_1 знову змінюється. Процедура повторюється доти, поки при русі по обох координатах функція F перестає збільшуватися.

Назва релаксаційного методу походить від слова релаксація, тобто розслаблення (ослаблення). Мається на увазі, що на кожному етапі «розслабляється» одна з координат і точка одержує можливість рухатися по цій координаті.

Релаксаційний метод пошуку дуже простий і його легко автоматизувати. Основний недолік методу полягає в тому, що число кроків у релаксаційній схемі сильно залежить від положення лінії рівня щодо координатних осей.

На рис. 4.6 показана одна і та сама поверхня у трьох різних положеннях відносно початку координат. У випадку, зображеному на рис. 4.6, а, пошук екстремуму закінчується за два кроки. У випадку на рис. 4.6, б число кроків дуже велике і теоретично може бути нескінченним. У випадку на рис. 4.6, в взагалі не попадаємо в точку екстремуму: рух із точки A_1 в обох напрямках призводить до зменшення функції F , хоча ця точка й не є точкою максимуму.

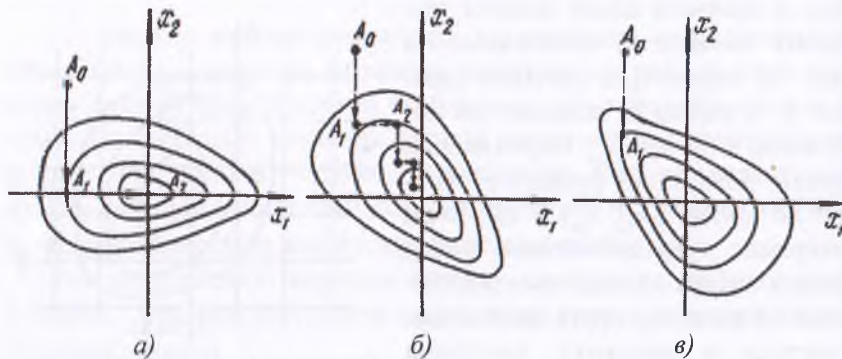


Рисунок 4.6 – Залежність кількості кроків від положення функції мети відносно осей координат (розворот функції)

Гradientний метод. Gradientом функції називається вектор, проєкціями якого на координатні осі є часткові похідні функції за координатами:

$$\text{grad } F = \left(\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots, \frac{\partial F}{\partial x_n} \right).$$

Напрямок градієнта – це напрямок найбільш швидкого зростання функції, найбільш «крутого схилу» поверхні функції мети (рис. 4.7, а). Проекція градієнта G на площину змінних x_1 і x_2 перпендикулярна до дотичної лінії рівня функції F (рис. 4.7, б). Напрямок вектора G визначається кутом α , складеним вектором G з додатним напрямком осі x_1 .

При пошуку екстремуму gradientним методом у початковій точці A_0 визначають похідні $\frac{\partial F}{\partial x_i}$, потім роблять крок у напрямку градієнта. Координати нової точки A_1 обчислюють за формулами:

$$x_{11} = x_{10} + h \frac{\partial F}{\partial x_1} / \sqrt{\left(\frac{\partial F}{\partial x_1} \right)^2 + \left(\frac{\partial F}{\partial x_2} \right)^2};$$

$$x_{12} = x_{20} + h \frac{\partial F}{\partial x_2} / \sqrt{\left(\frac{\partial F}{\partial x_1} \right)^2 + \left(\frac{\partial F}{\partial x_2} \right)^2}.$$

Розмір кроку вибирають досить малим. У точці A_1 знову обчислюють похідні і визначають новий напрямку градієнта. Кроку вибирають так, щоб при переході від однієї точки до іншої напрямку градієнта змінювався на $10 \dots 15^\circ$.

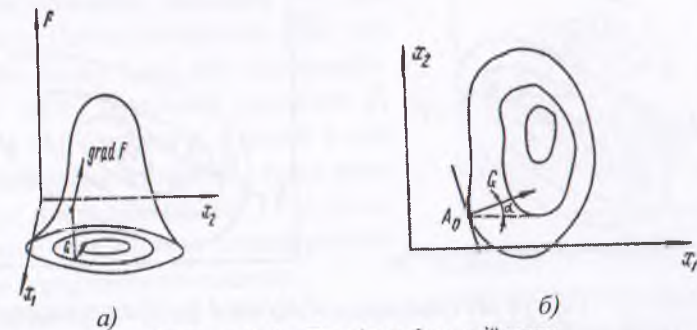


Рисунок 4.7 – Gradient функції мети

Переміщення триває доти, поки похідні $\frac{\partial F}{\partial x_1}$ та $\frac{\partial F}{\partial x_2}$ не стануть близькі до нуля (що говорить про досягнення стаціонарної точки) або поки не будуть досягнуті межі області задання змінних x_1, x_2 .

На рис. 4.8 показано траєкторію переміщення від початкової точки A_0 до максимуму функції F .

Метод найшвидшого підйому (найшвидшого спуску). Недолік градієнтного методу полягає в тому, що на кожному кроці треба обчислювати всі похідні функції F і визначати напрямок градієнта (при великому числі змінних x це трудомістка операція). Цього недоліку позбавлений метод найшвидшого підйому, також відомий за назвою «метод найшвидшого спуску», тому що він розроблявся для задач мінімізації.

Так само як у градієнтному методі, у початковій точці A_0 визначають напрямок градієнта і переміщуються в цьому напрямку. Однак переміщуються не на один крок, а на кілька кроків доти, поки функція F буде зростати. Коли функція F починає спадати (у точці A_1), як показано на рис. 4.9), знову обчислюють частинні похідні і визначають новий напрямок градієнта. На рис. 4.9 показана траєкторія переміщення до оптимуму. При використанні цього методу, як правило, забезпечується найшвидший пошук екстремуму.

Методи випадкового пошуку. Методи найшвидшого спуску, градієнтний і релаксацийний придатні для відшукування екстремуму тільки унімодалних функцій – функцій, що мають один максимум. Якщо ж максимумів декілька, то залежно від положення початкової точки ці методи приводять до тієї або іншої точки локального

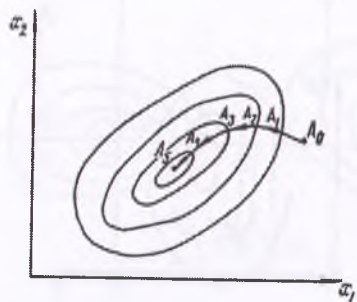


Рисунок 4.8 – Пошук екстремуму градієнтним методом

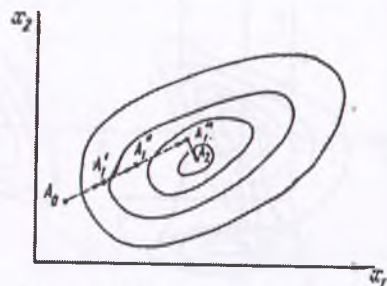


Рисунок 4.9 – Пошук екстремуму методом найшвидшого спуску

екстремуму, у той час як функція мети може мати найбільше значення (глобальний екстремум) зовсім в іншій точці.

На рис. 4.10 показана траєкторія пошуку з точки A_0 , що приводить до вершини B_1 , у той час як найбільшого значення функції F досягає у вершині B_4 . Тому всі ці методи мають назву локального пошуку.

Для відшукування глобального екстремуму можна скористатися методом повного перебору або яким-небудь варіантом випадкового пошуку. У найпростішому варіанті випадкового пошуку початкова точка A_0 вибирається випадково. У ній вимірюється значення функції F і це значення запам'ятовується. Наступна точка A_1 і всі наступні вибираються також зовсім випадково. Обмірюване в кожній точці нове значення функції F порівнюється з тим, що зберігається в пам'яті. Якщо нове значення більше старого, воно заноситься у пам'ять замість старого; якщо нове значення менше, воно відкидається.

Така проста схема випадкового пошуку застосовується рідко, звичайно використовуються більш складні модифікації. Так, наприклад, один з можливих варіантів – це випадковий пошук із проміжними підйомами. У цьому варіанті вибирають випадкову точку A_0 , роблять з неї підйом за градієнтом до найближчої локальної вершини B_0 і запам'ятовують значення F у цій вершині. Потім вибирають нову випадкову точку і від неї піднімаються до іншої локальної вершини B_1 (рис. 4.11). Значення функції в цій вершині запам'ятовують, якщо воно більше, ніж у попередній, і т. д. Кожний етап складається з випадкового кроку і градієнтного пошуку.

Порівняння. Різні методи пошуку мають свої переваги і недоліки.



Рисунок 4.10 – Поверхня з багатьма екстремумами

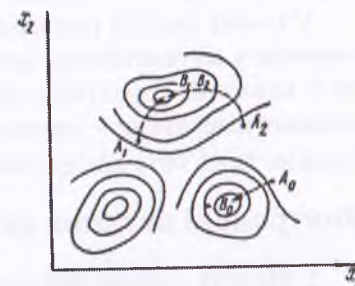


Рисунок 4.11 – Пошук екстремуму випадковим методом з підйомами за градієнтом

Вибір методу, у першу чергу, залежить від виду функції мети. Якщо заздалегідь відомо, що функція має один екстремум, можна використати один з локальних методів пошуку: релаксаційний, градієнтний або метод найшвидшого спуску. При цьому, якщо функція мети має вигляд

$$F(x_1, x_2, \dots, x_n) = f(x_1) + f(x_2) + \dots + f(x_n).$$

або мало відрізняється від такої функції, то для відшукування її екстремуму найкраще застосувати метод Гаусса. Якщо всі змінні тісно зв'язані й фізично однорідні, то найкраще застосовувати метод найшвидшого спуску. Поблизу оптимуму краще застосовувати градієнтний метод. При великій розмірності завдання найбільш ефективним є метод випадкового пошуку.

Якщо ж функція має декілька локальних екстремумів, необхідно застосовувати повний перебір або які-небудь варіанти випадкового пошуку.

Для функцій, що мають гострі круті гребені або западини, застосовуються у поєднанні випадковий пошук і градієнтний метод.

Дотепер вибір найбільш ефективного методу пошуку значною мірою визначається досвідом, умінням та інтуїцією дослідника. В останні роки активно розвивається метод пошуку екстремумів, заснований на генетичних алгоритмах. Він ефективний у тих випадках, коли про функції та її екстремуми немає зовсім ніякої інформації. У сполученні з описаними вище алгоритмами метод генетичних алгоритмів – найефективніший, проте потребує значних обчислювальних ресурсів і спеціалізованого програмного забезпечення.

У цьому розділі розглянуті особливості тільки деяких з безлічі можливих математичних методів теоретичних досліджень. Детальне їх вивчення і практичне застосування засноване на вивченні спеціальної літератури – залежно від конкретного наукового напрямку, завдань та об'єкта дослідження.

Контрольні питання до розділу 4

1. Назвіть основні завдання теоретичних досліджень.
2. Поясніть, що є метою теоретичних досліджень.
3. Для чого проводять розчленовування й об'єднання елементів досліджуваної системи (об'єкта, явища, процесу)?

4. Поясніть, що таке загальна теорія систем.
5. На яких постулатах ґрунтується загальна теорія систем?
6. Поясніть, що таке умови й вимоги науково-дослідного завдання.
7. Назвіть відомі Вам математичні методи наукових досліджень.
8. Для чого призначена математична модель досліджуваного об'єкта?
9. Поясніть, що таке динаміка процесу.
10. Чим відрізняється стаціонарність від нестаціонарності імовірнісних об'єктів?
11. У чому полягає принцип суперпозиції?
12. Поясніть, що розуміється під перехідним процесом.
13. Що таке статична характеристика об'єкта?
14. Для чого призначений принцип лінеаризації характеристик?
15. Для чого призначений контроль розмірності?
16. Для чого призначений контроль фізичного змісту виразу?
17. Для чого призначені аналітичні методи дослідження?
18. Розшифруйте аббревіатури ФЧХ, АФЧХ, ЛАЧХ.
19. Поясніть, що таке і для чого призначена цільова функція.
20. Які існують методи оптимізації?

Розділ 5

МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ ТЕХНІЧНИХ ОБ'ЄКТІВ, ПРИСТРОЇВ І СИСТЕМ

5.1 Моделювання як метод наукового пізнання

У наукових дослідженнях моделювання почало застосовуватися ще в далекій давнині й поступово захоплювало всі нові галузі наукових знань: технічне конструювання, будівництво й архітектуру, астрономію, фізику, хімію, біологію, суспільні науки. Великі успіхи й визнання, практично у всіх галузях сучасної науки приніс моделюванню минулий вік. Однак, методологія моделювання довгий час розвивалася окремими науками незалежними шляхами, з відсутністю єдиної системи понять і термінології. Лише поступово вченими й інженерами усвідомлювалася роль моделювання як універсального методу наукового пізнання.

Зараз термін «модель» широко використовується у будь-яких сферах людської діяльності й має безліч значень. Із цього різноманіття розглянемо тільки такі моделі, які є інструментами одержання нових знань. Формування поняття «модель», загалом, неоднозначне і далі буде поетапно доповнюватися й уточнюватися.

Отже, вважаємо, що модель – це такий матеріальний об'єкт, або об'єкт, який уявляється і заміщає в процесі дослідження вихідний об'єкт-оригінал так, щоб безпосереднє вивчення «моделі» дало нові знання про об'єкт-оригінал.

Під моделюванням будемо розуміти процес побудови, вивчення й застосування моделей. Процес моделювання тісно пов'язаний з такими категоріями, як абстракція, аналогія, гіпотеза тощо. Моделювання обов'язково включає в себе і побудову абстракцій та умовиводу за аналогією, і конструювання наукових гіпотез. Безумовно, мається на увазі, що інженер-дослідник у достатній мірі знає фізику процесів, що відбуваються у модельованому пристрої або системі,

і здатен формалізувати завдання для своєї предметної області. Головна особливість моделювання полягає в тому, що це метод опосередкованого пізнання за допомогою об'єктів-заступників. Модель завжди виступає як своєрідний, пристосований до розв'язання конкретного завдання, інструмент пізнання. Цей інструмент дослідник ставить між собою та об'єктом, за його допомогою вивчає той об'єкт, який його цікавить. Зазначена особливість методу моделювання визначає специфічні форми використання абстракцій, аналогій, гіпотез, інших категорій і методів пізнання.

Необхідність використання методів моделювання визначається тим, що багато об'єктів (або проблем, що висуюються до цих об'єктів) безпосередньо досліджувати або зовсім неможливо, або ж це дослідження вимагає неприпустимих витрат часу й засобів. Процес моделювання включає в себе три складові:

- 1) суб'єкт (дослідник);
- 2) об'єкт дослідження;
- 3) модель, що опосередковує відносини суб'єкта, що пізнає, і пізнаваного об'єкта пізнання.

Нехай є або необхідно створити деякий об'єкт (пристрій) А. Ми конструємо (матеріально або подумки) або знаходимо в реальному світі інший об'єкт-заступник В – модель об'єкта А.

Перший етап побудови моделі припускає наявність певних, бажано однозначних, знань про об'єкт-оригінал. Пізнавальні можливості моделі обумовлені тим, що модель відображає які-небудь, встановлені дослідником, суттєві риси об'єкта-оригіналу. Питання про необхідність і достатню міру подібності оригіналу й моделі вимагає конкретного, додаткового аналізу. Очевидно, модель втрачає свою суть як у випадку повної тотожності з оригіналом (тоді вона стає оригіналом), так і у випадку надмірної (у всіх істотних значеннях) відмінності від оригіналу.

Таким чином, вивчення одних сторін модельованого об'єкта (пристрою, системи) здійснюється ціною відмови від урахування інших сторін. Тому будь-яка модель заміщає оригінал лише в строго обмеженому, необхідному для одержання конкретного результату, змісті. Із цього випливає, що для одного об'єкта може бути побудовано багато «спеціалізованих» моделей, які концентрують увагу на певних сторонах досліджуваного об'єкта або ж характеризують об'єкт із різним ступенем деталізації.

На *другому* етапі процесу моделювання модель виступає самостійним об'єктом дослідження. Одна з форм такого дослідження – проведення «модельних» експериментів. Тут свідомо змінюються умови функціонування моделі й систематизуються дані щодо її «поводження» в нових умовах. Проведення «модельних» експериментів дозволяє досліджувати реакцію об'єкта при неприпустимих у реальному світі умовах, наприклад, в аварійних режимах експлуатації. Можна досліджувати модель і при відхиленнях від номінальних умов експлуатації об'єкта, наприклад, припустимим є збільшення опору обмоток двигуна при перегріві, змінах моментів інерції механізму й інших (такі експерименти описуються теорією чутливості). Кінцевим результатом цього етапу є безліч знань R про моделі.

На *третьому* етапі здійснюється перенесення знань, отриманих за допомогою моделі, на оригінал – формування безлічі знань S про об'єкт. Процес переносу знань проводиться за певними правилами. Знання про моделі повинні бути скореговані з урахуванням тих властивостей об'єкта-оригіналу, які не знайшли відображення або були змінені при побудові моделі. Тоді можна, з достатньою впевненістю, переносити будь-який результат від моделі до оригіналу (якщо цей результат пов'язаний ознаками подібності оригіналу й моделі). Якщо ж певний результат модельного дослідження пов'язаний з відмінністю моделі від оригіналу, то цей результат переносити неправомірно.

Четвертий етап – практична перевірка одержуваних за допомогою моделей знань та їхнє використання для побудови узагальнюючої теорії об'єкта, його перетворення або керування ним.

Для розуміння сутності моделювання важливо не випустити з уваги те, що моделювання – не єдине джерело знань про об'єкт. Процес моделювання «занурений» у більш загальний процес пізнання. Ця обставина враховується не тільки на етапі побудови моделі, але і на завершальній стадії, коли відбувається об'єднання й узагальнення результатів дослідження, одержуваних на основі різноманітних засобів пізнання.

У цілому, моделювання – циклічний процес. Це означає, що за першим чотирьохетапним циклом може піти другий, третій і т. д. У ході моделювання знання про досліджуваний об'єкт розширюються й уточнюються, а вихідна модель поступово вдосконалюється. Недоліки, виявлені після першого циклу моделювання, обумовлені

низьким знанням властивостей об'єкта, неправильною формалізацією опису об'єкта і помилками в побудові моделі. Наступними циклами можна виправити модель. Таким чином, у методології моделювання закладені великі можливості саморозвитку.

5.2 Особливості застосування методу математичного моделювання

Більшість об'єктів, досліджуваних у електромеханіці, судноводінні, теплоенергетиці та багатьох інших технічних науках, може бути охарактеризована кібернетичним поняттям «складна система». Наприклад, практично всі сучасні електромеханічні пристрої є «складними системами», і тому їх можна віднести до «складних пристроїв». Очевидно, що будь-який поділ на «просте» й «складне» досить умовний. Але завжди варто пам'ятати, що глибоке проникнення математики в техніку пов'язане з подоланням тільки суб'єктивних труднощів. У цьому «винна» і сама математика, яка бурхливо розвивається протягом останніх століть і дає безліч оригінальних аналітичних і числових методів, найчастіше доволі складних для практичного застосування.

Найпоширеніше розуміння системи, як певної сукупності елементів, що перебувають у взаємодії й утворюють деяку цілісність, єдність – «складний технічний пристрій».

Важливою ознакою будь-якої системи є емерджентність – наявність таких рис, які не властиві жодному з елементів, що входять у систему. Тому при вивченні складних систем часто недостатньо користуватися методом їхнього розчленовування на елементи з наступним вивченням цих елементів окремо. Певні труднощі таких досліджень полягають у тому, що майже не існує об'єктів і пристроїв, які можна було б розглядати як окремі (позасистемні) елементи.

Хибність системи визначається кількістю вхідних у неї елементів, зв'язками між цими елементами, а також взаєминами між системою і середовищем. Наприклад, сучасний електропривод має всі ознаки складної системи – він поєднує величезне число елементів, відрізняється різноманітністю внутрішніх зв'язків і зв'язків з іншими системами: енергетичною, керуючою, зовнішньої навколишньої (на електропривод діють зовнішні збурення і технологічні процеси, об'єктивні й суб'єктивні фактори та безліч інших).

Проте моделювати можна об'єкт будь-якої природи і складності (зовсім інше питання, яким чином, засобами і з якою точністю). І саме складні об'єкти й системи становлять найбільший інтерес для моделювання; саме тут моделювання може дати результати, які не можна одержати іншими способами дослідження. Потенційна можливість математичного моделювання будь-яких пристроїв і процесів не означає її успішної реалізації при даному рівні математичних і фізико-технічних знань, наявної конкретної інформації та обчислювальної техніки. І, хоча не можна вказати абсолютні межі математичного формалізму технічних завдань, завжди будуть існувати ще неформалізовані проблеми, а також ситуації, де математичне моделювання недостатньо ефективне.

5.3 Особливості спостережень і вимірів при розробці моделі

Вже тривалий час головним гальмом практичного застосування математичного моделювання є наповнення розроблених моделей конкретною та якісною інформацією. Точність і повнота первинної інформації, реальні можливості її збору й обробки багато в чому визначають вибір типів прикладних моделей. З іншого боку, результати досліджень методів моделювання висувають все нові й нові вимоги до системної та апіорної інформації. Однак це завдання ідентифікації (див. розд. 7) – науки, основною метою якої є створення підґрунтя математичних моделей, як правило, на основі експериментальних досліджень. Можна помітити, що моделювання й ідентифікація взаємозалежні і взаємопов'язані, багато в чому визначають послідовність створення й дослідження моделей.

Залежно від модельованих об'єктів і призначення моделей, використовувана в них вихідна інформація має істотно різний характер і походження. Вона може бути розділена на дві категорії:

- про минулий розвиток і сучасний стан об'єктів (спостереження та їхня обробка);
- про майбутній розвиток об'єктів, включаючи дані про очікувані зміни їхніх внутрішніх параметрів і зовнішніх умов (прогнози).

Друга категорія інформації є результатом самостійних досліджень, які також можуть виконуватися за допомогою моделювання.

Методи спостережень і використання результатів цих спостережень розробляються статистикою. Зараз варто відзначити тільки

специфічні проблеми спостережень, пов'язані з моделюванням процесів у різних системах і пристроях.

Низка процесів є масовими. Вони характеризуються закономірностями, що не виявляють себе на підставі лише одного або декількох спостережень. Тому моделювання може опиратися на масові спостереження.

Інша проблема породжується динамікою процесів, мінливістю їхніх параметрів і структурних відносин (зв'язків). Внаслідок цього такі процеси доводиться постійно тримати під спостереженням, необхідно мати стійкий потік нових даних. Іноді спостереження за реальними динамічними процесами, з наступною обробкою емпіричних даних, можуть займати досить багато часу, тому при побудові математичних моделей потрібно коректувати вихідну інформацію, відкидати несуттєву, тобто фільтрувати.

Пізнання кількісних відносин опирається на виміри. Точність вимірювань, у значній мірі, визначає й точність кінцевих результатів кількісного аналізу за допомогою моделювання. Тому необхідною умовою ефективного використання математичного моделювання є вдосконалення методів вимірів. Застосування математичного моделювання загострило проблему вимірів, кількісних зіставлень різних аспектів та явищ, що відбуваються у пристроях і системах, вірогідності й повноти одержуваних даних, їхнього захисту від технічних перекручувань.

У процесі моделювання виникає взаємодія «первинних» і «вторинних» змінних параметрів. Будь-яка модель опирається на певну систему змінних параметрів (швидкість, напруга, температура, витрата і т. д.). У той же час, одним з важливих результатів моделювання є одержання нових (вторинних) інтегрованих або невимірюваних параметрів (показників, вимірників) – ефективності, якості і багатьох інших. Ці змінні параметри можуть зазнавати впливу від недостатньо обґрунтованих первинних параметрів, що змушує розробляти особливу методику коректування первинних параметрів для вихідних моделей.

5.4 Випадковість і невизначеність

Розрізняють два типи невизначеності: «щиру», обумовлену властивостями досліджуваних процесів, і «інформаційну», пов'язану з неповнотою й неточністю наявної інформації про ці

процеси. Щиру невизначеність не можна змішувати з об'єктивним існуванням різних режимів функціонування пристроїв і можливістю свідомого вибору серед них ефективних варіантів. Мова йде про принципову неможливість точного вибору єдиного (оптимального за яким-небудь із прийнятих критеріїв) варіанта.

Невизначеність викликається двома основними причинами. По-перше, хід процесів, а також зовнішні впливи на ці процеси не можуть бути точно передбачувані через дію випадкових факторів й обмеженості людського пізнання в кожен момент.

По-друге, керування технічним пристроєм або системою не всесильне, а наявність безлічі самостійних додаткових факторів не дозволяє точно передбачити результати їхніх взаємодій. Неповнота й неточність інформації про об'єктивні процеси підсилюють щиру невизначеність. На перших етапах досліджень з моделювання технічних пристроїв і систем (електромеханіка, теплоенергетика та ін.), застосовуються, в основному, моделі *детерміністського* типу. Це поняття походить від латинського слова *determino* – визначаю (філософське вивчення закономірного взаємозв'язку й причинної обумовленості всіх явищ; протистоїть індетермінізму, що заперечує загальний характер причинності). У детермінованих моделях усі параметри передбачаються точно відомими. Однак детерміністські моделі неправильно розуміти в механістичному дусі, ототожнювати їх з моделями, які позбавлені всіх «ступенів вільності» (можливостей вибору) і мають єдине припустиме і вірне рішення. Будь-який формалізований й однозначний опис, наприклад, рівняння руху електропривода, рівняння теплової рівноваги, є детермінованим.

У результаті накопичення досвіду використання строго детерміністських моделей були створені реальні можливості успішного застосування більш розвинутої методології моделювання процесів в електромеханічних, теплових і багатьох інших технічних пристроях, що враховують стохастичну й невизначеність (стохастичний – від грецького *stochastikos* – що вміє вгадувати, випадковий, імовірнісний). Тут можна виділити два основних напрямки досліджень.

По-перше, удосконалення методики використання моделей строго детерміністського типу: проведення різноманітних розрахунків і модельних експериментів з варіацією конструкції моделі та її вихідних даних; вивчення стійкості і надійності одержуваних розв'язків, виділення зони невизначеності; включення в модель резервів, за-

стосування прийомів, що підвищують пристосовність розв'язків до ймовірних і непередбачених ситуацій, тощо.

По-друге, поширення моделей, які безпосередньо відображають стохастизм і невизначеність процесів і використовують відповідний математичний апарат: теорію ймовірностей і математичну статистику, нейросіткове моделювання й нечітку логіку, теорію статистичних розв'язків, стохастичне програмування, теорію випадкових процесів і багато інших методів.

5.5 Адекватність моделі

Складність електромеханічних, теплових та інших процесів і явищ, інші зазначені вище особливості технічних пристроїв і систем, ускладнюють не тільки побудову математичних моделей, але й перевірку їхньої адекватності – оцінку істинності одержуваних результатів.

Достатньою умовою істинності результатів моделювання, як і будь-яких інших форм пізнання, є збіг результатів дослідження зі спостережуваними фактами – експериментами. Категорія «практика» збігається з категорією «дійсність». Принцип «практика – критерій істини» більшою мірою застосований до простих описативних моделей, які використовуються для пасивного опису і пояснення дійсності.

Головне завдання й електромеханіки, і теплоенергетики, і будь-якої іншої інженерної науки, конструктивне: розробка науково обґрунтованих методів керування технічною системою або пристроєм. Тому розповсюдженим є тип математичних моделей – моделі керування і регульованих об'єктів, процесів, пристроїв, призначених для перетворення дійсності. Такі моделі іноді називають нормативними. Але, якщо орієнтувати нормативні моделі тільки на підтвердження дійсності, то вони не зможуть служити інструментом розв'язання якісно нових соціально-економічних, технічних і наукових завдань.

Специфіка верифікації моделей полягає в тому, що вони, як правило, «конкурують» з іншими, що вже знайшли практичне застосування, методами. Термін «верифікація» виник від латинських слів *verus* – щирий і *facio* – роблю, перевіряю. Отже, верифікація – емпіричне підтвердження теоретичних положень науки шляхом зіставлення їх з отриманими експериментально. При цьому, далеко не

завжди можна поставити чистий експеримент з верифікації моделі, усунувши вплив інших керуючих дій і збурюючи вплив на модельований об'єкт.

Ситуація ще більше погіршується, коли ставиться питання про верифікацію складних моделей (як дескриптивних, так і нормативних): часто складні математичні моделі мають тисячі логічних розгалужень, складних зв'язків, і важко перебрати всі можливі комбінації, щоб перевірити правильність функціонування моделі у всьому різноманітті режимів її роботи.

Незважаючи на зазначені ускладнюючі обставини, відповідність моделі фактам і тенденціям реального життя залишається найважливішим критерієм, що визначає напрямки вдосконалювання моделей. Всебічний аналіз розбіжностей, які виявляються, між дійсністю й моделлю, зіставлення результатів по моделі з результатами, отриманими іншими методами, допомагають виробити шляхи корекції моделей.

Значна роль у перевірці моделей належить логічному аналізу, у тому числі засобами самого математичного моделювання. Такі формалізовані прийоми верифікації моделей, як доведення існування розв'язку в моделі, перевірка істинності статистичних гіпотез про зв'язки між параметрами й змінними моделі, зіставлення розмірності величин тощо, дозволяють звузити клас потенційно «неправильних» моделей.

Внутрішня несуперечність передумов моделі перевіряється також шляхом порівняння одержуваних з її допомогою наслідків, а також з наслідками «конкуруючих» моделей.

Оцінюючи сучасний стан проблеми адекватності математичних моделей, варто визнати, що створення конструктивної комплексної методики верифікації моделей, яка враховує як об'єктивні особливості модельованих об'єктів, так й особливості їхнього пізнання, як і раніше, є одним з найбільш актуальних завдань сучасних досліджень.

5.6 Загальна класифікація математичних моделей

За цільовим призначенням математичні моделі поділяються на теоретично-аналітичні, які використовуються в дослідженнях загальних властивостей і закономірностей процесів, і прикладні, які застосовуються при розв'язанні конкретних завдань (моделі аналізу, оптимізації, регулювання й керування).

Математичні моделі можуть призначатися для дослідження різних сторін системи або пристрою і для їх окремих частин. При класифікації моделей за досліджувальними процесами та змістовою проблематикою можна виділити моделі в цілому й моделі підсистем.

Зупинимося на характеристиці таких класів математичних моделей, з якими пов'язані найбільші особливості методології і техніки моделювання.

Відповідно до загальної класифікації математичних моделей вони поділяються на функціональні і структурні, а також включають в себе проміжні форми (структурно-функціональні). У дослідженнях багатьох технічних пристроїв і систем, зокрема автоматичних пристроїв і систем керування, дуже часто застосовуються структурні моделі, оскільки для керування й регулювання дуже велике значення мають взаємозв'язки підсистем. Типовими структурними моделями безлічі пристроїв є моделі з від'ємними зворотними зв'язками. Функціональні й структурні моделі широко застосовуються в описі систем регулювання, коли на поведінку об'єкта («вихід») впливають шляхом зміни «входу». Прикладом може служити модель електропривода керованого компресора. Той самий об'єкт може описуватися одночасно і структурною, і функціональною моделлю. Не можна забувати про те, що принципова схема пристрою, системи або об'єкта теж є моделлю, представленою у вигляді символів і позначень – своєрідною знаковою моделлю.

Вище вже показувалися розходження між моделями дескриптивними й нормативними. Дескриптивні моделі відповідають на запитання: «Як це відбувається?» або «Як це найімовірніше може далі розвиватися?». Вони тільки пояснюють спостережувані факти або дають імовірний прогноз. Нормативні моделі відповідають на запитання: «Як це повинно бути?», тобто припускають цілеспрямовану діяльність. Типовим прикладом нормативних моделей є моделі оптимального керування, які формалізують тим або іншим способом мету керування, можливості й засоби її досягнення.

Застосування дескриптивного підходу в моделюванні мети викликане необхідністю емпіричного виявлення різних залежностей у технічних системах і пристроях, установлення статистичних закономірностей, вивчення ймовірних шляхів розвитку яких-небудь процесів при умовах, що не змінюються або проходять без зовнішніх впливів.

Чи є математична модель дескриптивною або нормативною, залежить не тільки від її математичної структури, але і від характеру використання цієї моделі. Наприклад, модель електричної машини дескриптивна, якщо вона використовується для аналізу сталих режимів. Але ця ж математична модель стає нормативною, коли застосовується для розрахунків збалансованих варіантів оптимального керування верстатом, наприклад, за критерієм мінімальної собівартості продукції або максимальної продуктивності.

Багато математичних моделей сполучають ознаки дескриптивних і нормативних моделей. Типова ситуація, коли нормативна модель складної структури поєднує окремі блоки, які є окремими дескриптивними моделями. Дескриптивний підхід широко застосовується в *імітаційному моделюванні*.

За характером відображення причинно-наслідкових зв'язків розрізняють моделі строго детерміністські і моделі, що враховують випадковість і невизначеність. Необхідно розрізняти невизначеність, описувану ймовірнісними законами, і невизначеність, для опису якої закони теорії ймовірностей незастосовні. Другий тип невизначеності набагато складніший для моделювання.

За способом відображення фактору часу математичні моделі поділяються на статичні й динамічні. У статичних моделях всі залежності належать до одного моменту або періоду часу. Динамічні моделі характеризують зміни процесів у часі. Моделі розрізняються також за тривалістю розглянутого періоду часу. Сам час у математичних моделях може змінюватися або безупинно, або дискретно.

Моделі динамічних і статичних процесів надзвичайно різноманітні за формою математичних залежностей. Особливо важливо виділити клас лінійних моделей, найбільш зручних для аналізу й обчислень, які внаслідок цього мають велике поширення. Розходження між лінійними та нелінійними моделями суттєві не тільки з математичної точки зору, але й у теоретичному відношенні, оскільки багато залежностей у механіці, електромеханіці, електроніці носять принципово нелінійний характер: насичення операційних підсилювачів, сили тертя, дискретність перетворень та ін. Методи «лінійної» теорії автоматичного керування (ТАК) істотно відрізняються від методів «нелінійної» ТАК. Від того, чи передбачаються безлічі можливостей підсистем опуклими або ж неопуклими, істотно залежать висновки про можливість сполучення методів корекції та керування.

За співвідношенням екзогенних (зовнішнього походження) і ендогенних (внутрішнього походження) змінних, що входять у модель, вони можуть поділятися на відкриті і закриті. Повністю відкритих моделей не існує; модель повинна містити хоча б одну ендогенну змінну. Повністю закриті математичні моделі, тобто до яких не входять екзогенні змінні, винятково рідкі. Їхня побудова вимагає повного абстрагування від «середовища» – серйозного огрубіння реальних систем, що завжди мають зовнішні зв'язки. Більшість математичних моделей займає проміжне положення й розрізняється за ступенем відкритості (закритості).

Моделі функціонального рівня можна розділити на агреговані й деталізовані. Наприклад, залежно від того, включають моделі просторові фактори й умови або не включають, розрізняють моделі просторові і точкові.

Отже, загальна класифікація математичних моделей включає в себе більше десятка ознак. З розвитком математичних досліджень проблема класифікації застосовуваних моделей ускладнюється. Поряд з появою нових типів моделей, особливо змішаних типів, нових ознак їхньої класифікації, здійснюється процес інтеграції моделей різних типів у більш складні модельні конструкції.

5.7 Основні етапи математичного моделювання

Деякі етапи процесу моделювання вже розглядалися вище. У різних галузях знань вони набувають своїх специфічних рис. Проаналізуємо послідовність і зміст етапів одного циклу математичного моделювання механічних й електромеханічних пристроїв.

5.7.1 Постановка проблеми та її якісний аналіз

Головне тут – чітко сформулювати сутність проблеми, прийняті припущення й ті питання, на які потрібно одержати відповіді. Цей етап включає в себе виділення найважливіших рис і властивостей модельованого об'єкта, абстрагування від другорядних факторів; вивчення структури об'єкта й основних залежностей, що зв'язують його елементи; формулювання гіпотез (хоча б попередніх), які пояснюють поведінку й розвиток об'єкта.

5.7.2 Побудова математичної моделі

Це етап формалізації проблеми, вираження її у вигляді конкретних математичних залежностей і відношень (функцій, рівнянь, нерівностей і т. д.). Зазвичай спочатку визначається основна конструкція (тип) математичної моделі, а потім уточнюються деталі цієї конструкції (конкретний перелік змінних і параметрів, форма зв'язків). Таким чином, побудова моделі поділяється на декілька стадій.

Неправильно думати, що чим більше фактів враховує модель, тим вона краще «працює» і дає кращі результати. Те ж можна сказати про такі характеристики складності моделі, як використання форми математичних залежностей (лінійні та нелінійні), врахування факторів випадковості й невизначеності та ін. Зайва складність і громіздкість моделі ускладнюють процес дослідження. Потрібно враховувати не тільки реальні можливості інформаційного й математичного забезпечення, але і зіставляти витрати на моделювання з одержуваним ефектом (при зростанні складності моделі приріст витрат може перевищити приріст ефекту).

Одна з важливих особливостей математичних моделей – потенційна можливість їхнього використання для вирішення проблем різного характеру. Тому, навіть зіштовхуючись з новим технічним завданням, не потрібно намагатися «винаходити» модель; спочатку необхідно спробувати застосувати для розв'язання цього завдання вже відомі моделі. Тільки в тому випадку, якщо відомих розв'язків немає (що досить мало ймовірно), варто зайнятися новою розробкою.

У процесі побудови моделі здійснюється взаємне зіставлення двох систем наукових знань – фізико-технічних і математичних. Це природне прагнення до того, щоб одержати модель, яка належить до добре вивченого класу математичних задач. Часто це вдається зробити шляхом деякого спрощення вихідних передумов моделі, що не спотворюють істотних рис модельованого об'єкта. Однак, часто виникає і така ситуація, коли формалізація проблеми приводить до невідомої раніше математичної структури. Потреби науки й практики всередині минулого століття сприяли розвитку математичного програмування, теорії ігор, функціонального аналізу, обчислювальної математики й числових методів. Уже видно, що розвиток інженерно-технічних наук став важливим стимулом для створення нових розділів математики.

5.7.3 Математичний аналіз моделі

Метою цього етапу є з'ясування загальних властивостей моделі. Тут застосовуються чисто математичні прийоми дослідження. Найбільш важливий момент – доведення існування розв'язків у сформульованій моделі (теорема існування). Якщо вдасться довести, що математичне завдання не має розв'язку, то необхідність у наступній роботі щодо первісного варіанта моделі відпадає; варто скорегувати або постановку технічного завдання, або способи його математичної формалізації. При аналітичному дослідженні моделі з'ясовуються такі питання, як, наприклад, чи існує один єдиний розв'язок, які змінні (невідомі) можуть входити до розв'язку, які співвідношення будуть між ними, у яких межах і залежно від яких вихідних умов вони змінюються, які тенденції їхньої зміни і т. д.

Аналітичне дослідження моделі в порівнянні з емпіричним (числовим) має ті переваги, що одержувані висновки зберігають свою силу при різних конкретних значеннях зовнішніх і внутрішніх параметрів моделі. Але варто пам'ятати, що складні аналітичні вирази, «багатоповерхові» формули необхідно програмувати для розрахунку за допомогою ЕОМ, а наочність і зручність використання таких складних викладень, найчастіше, досить спірна. Знання загальних властивостей моделі має настільки важливе значення, що часто заради доведення подібних властивостей дослідники свідомо йдуть на ідеалізацію первісної моделі. І все-таки, моделі складних технічних об'єктів на превелику силу піддаються аналітичному дослідженню й опису. Наприклад, навіть одержавши такий аналітичний «багатоповерховий» опис, дослідник часто не в змозі провести його детальний аналіз. У тих випадках, коли аналітичними методами не вдається з'ясувати загальні властивості моделі, а спрощення моделі приводить до неприпустимих результатів, переходять до числових методів дослідження за допомогою ЕОМ. Цей перехід у наш час здійснюється повсюдно і продовжує бурхливо розвиватися у всіх галузях діяльності людства.

5.7.4 Підготовка вихідної інформації

Моделювання висуває тверді вимоги до системи інформації. У той же час реальні можливості одержання інформації обмежують вибір моделей, що призначаються для практичного використання.

Тут береться до уваги не тільки принципова можливість підготовки інформації (за певні строки), але й витрати на підготовку відповідних інформаційних масивів. Ці витрати не повинні перевищувати ефект від використання додаткової інформації.

У процесі підготовки інформації широко використовують методи ідентифікації, теорії ймовірностей, теоретичної та математичної статистики. При системному математичному моделюванні вихідна інформація, що використовуються в одних моделях, є результатом функціонування інших моделей.

5.7.5 Числове розв'язання

Цей етап включає в себе розробку алгоритмів для числового розв'язання завдання, складання програм на ЕОМ і безпосереднє проведення розрахунків. Труднощі цього етапу обумовлені, насамперед, суб'єктивними проблемами (знання специфічних розділів електроніки, механіки, термодинаміки, методів розрахунку, мов програмування, математики тощо), а також великою розмірністю задач.

Звичайно, розрахунки за допомогою математичної моделі носять різноманітний характер. Завдяки високій швидкодії сучасних ЕОМ вдається проводити числові «модельні» експерименти, вивчаючи «поводження» моделі при різних змінах деяких умов. Дослідження, проведені числовими методами, можуть суттєво доповнити, а часто й більш ефективно замінити, результати аналітичного дослідження. Для багатьох технічних моделей дослідження числовими методами є єдино здійсненними. Клас завдань, розв'язуваних числовими методами, значно ширший, ніж клас завдань, доступних аналітичному дослідженню.

5.7.6 Аналіз числових результатів та їх застосування

На цьому заключному етапі циклу постає питання про правильність і повноту результатів моделювання, а також про ступінь їх практичної застосованості.

Математичні методи перевірки (верифікації) можуть виявляти некоректні побудови моделі і тим самим звужувати клас потенційно правильних моделей. Неформальний аналіз теоретичних висновків і числових результатів, одержуваних за допомогою моделі, зістав-

лення їх з наявними знаннями й фактами дійсності також дозволяють виявляти недоліки постановки завдання, сконструйованої математичної моделі, її інформаційного та математичного забезпечення.

5.8 Взаємозв'язок етапів моделювання

Подивимось на зворотні зв'язки етапів, які виникають внаслідок того, що в процесі дослідження виявляються недоліки попередніх етапів моделювання. Вже на етапі побудови моделі може з'ясуватися, що постановка завдання суперечлива або приводить до занадто складної математичної моделі. Тоді вихідна постановка завдання коректується. Далі математичний аналіз моделі може показати, що невелика модифікація постановки завдання або її формалізації дає цікавий результат.

Найбільш часто необхідність повернення до попередніх етапів моделювання виникає при підготовці вихідної інформації. Може виявитися, що необхідна інформація відсутня або ж витрати на її підготовку занадто великі. Тоді доводиться повертатися до постановки завдання і її формалізації, змінюючи їх так, щоб пристосуватися до наявної інформації.

Оскільки математичні завдання можуть бути складні за своєю структурою, мати більшу розмірність, то часто трапляється, що відомі алгоритми й програми для ЕОМ не дозволяють вирішити завдання в первісному вигляді. Якщо неможливо в короткий строк розробити нові алгоритми й програми, вихідну постановку завдання й модель спрощують: знімають і поєднують умови, зменшують число факторів, нелінійні співвідношення замінюють лінійними, підсилюють детермінізм моделі і т. д..

Недоліки, які не вдається виправити на проміжних етапах моделювання, усуваються в наступних циклах, але результати кожного циклу мають і цілком самостійне значення. Почавши дослідження з побудови простої моделі, можна швидко одержати корисні результати, і на основі їхнього аналізу перейти до створення більш розробленої моделі, що доповнюється новими умовами, які вже включають уточнені математичні залежності.

По мірі розвитку й ускладнення математичного моделювання його окремі етапи відокремлюються в спеціалізовані галузі досліджень, підсилюється розходження між теоретико-аналітичними й

прикладними моделями, відбувається диференціація моделей по рівнях абстракції та ідеалізації.

Часто моделі втрачають безпосередній зв'язок з реальністю: вони мають справу з винятково ідеалізованими об'єктами і ситуаціями. При побудові таких моделей головним принципом є не стільки наближення до реальності, скільки одержання як можна більшого числа числових (або аналітичних) результатів за допомогою математичних доведень. Цінність цих моделей полягає у тому, що вони є теоретичною базою для побудови моделей прикладного типу.

Досить самостійними галузями досліджень стають підготовка й обробка інформації, розробка математичного забезпечення технічних завдань (створення баз даних і банків інформації, програм автоматизованої побудови моделей і програмного сервісу для користувачів). На етапі практичного використання моделей провідну роль повинні відігравати фахівці у відповідній галузі, головною ділянкою роботи яких залишається постановка і формалізація завдань, а також синтез процесу математичного моделювання.

5.9 Роль математичного моделювання у прикладних дослідженнях

Можна виділити, принаймні, чотири аспекти застосування математичних методів у вирішенні практичних проблем.

1. Удосконалювання системи інформації. Математичні методи дозволяють упорядкувати систему інформації, виявляти недоліки в наявній інформації й виробляти вимоги для підготовки нової інформації або її коректування. Розробка і застосування математичних моделей вказують шляхи вдосконалювання апріорної інформації, орієнтованої на вирішення певної системи завдань оптимізації та керування. Прогрес в інформаційному забезпеченні потреб електромеханіки опирається на бурхливий розвиток технічних і програмних засобів інформатики.

2. Інтенсифікація та підвищення точності розрахунків. Формалізація завдань і застосування ЕОМ багаторазово прискорюють типові, масові розрахунки, підвищують точність і скорочують трудомісткість, дозволяють проводити різноманітні техніко-економічні обґрунтування складних пристроїв і систем, які є недоступними при пануванні «ручної» технології.

3. Поглиблення кількісного аналізу проблем. Завдяки застосуванню методів моделювання значно підсилюються можливості конкретного кількісного аналізу; вивчення багатьох факторів, що роблять суттєвий вплив на статичні й динамічні процеси, кількісну оцінку наслідків зміни умов функціонування об'єктів.

4. Розв'язання принципово нових завдань. За допомогою математичного моделювання вдається вирішувати такі завдання, які іншими засобами вирішити практично неможливо, наприклад: знаходження оптимального варіанта експлуатації устаткування, імітація аварійних ситуацій, автоматизація контролю функціонування складних об'єктів і безліч інших.

Сфера практичного застосування методу моделювання обмежується можливостями й ефективністю формалізації проблем і ситуацій, а також станом інформаційного, математичного, технічного забезпечення використовуваних моделей. Прагнення у будь-якому випадку застосувати математичну модель може не дати гарних результатів через відсутність хоча б деяких необхідних умов.

Відповідно до сучасних наукових уявлень, системи ідентифікації, моделювання, розробки й прийняття рішень повинні сполучати формальні і неформальні методи, що взаємно підсилюють і взаємодоповнюють одне одного. Формальні методи є, насамперед, засобом науково обґрунтованої підготовки матеріалу для дій людини в процесах керування. Це дозволяє продуктивно використовувати досвід та інтуїцію людини, її вміння вирішувати погано формалізовані завдання.

Комп'ютерні технології в моделюванні різноманітних систем мають на увазі максимальне використання засобів, надаваних сучасними програмними продуктами. Найбільш зручним при моделюванні й автоматизованому проектуванні різноманітних механічних, електромеханічних, електронних та інших технічних систем є програмний продукт *Matlab* фірми *MathWorks, Inc.* У ньому є найширші можливості застосування віртуальних блоків силової та керуючої частини частини електроприводів, механічних елементів, електронних компонентів, елементів і систем гідравліки, механіки тощо. Віртуальні системи та об'єкти можуть бути набрані з віртуальних блоків окремих бібліотек *Matlab* і досліджені, тобто визначені їх статичні, енергетичні й динамічні характеристики. З отриманих характеристик можна судити про відповідність технічному завданню

на проектування, зробити наукові висновки, провести віртуальні експерименти з моделлю. Однак не завжди використання названої «віртуальної лабораторії» повністю задовольняє дослідника.

По-перше, дослідник може зіштовхнутися з елементами, яких немає у віртуальних бібліотеках.

По-друге, при дослідженні досить складних систем можна зіштовхнутися з серйозними складнощами обчислювального характеру, коли зібрана віртуальна модель не працює.

По-третє, на початковому етапі моделювання дослідника можуть цілком задовольнити результати наближені, які скоріше є граничними (еталонними), ніж реальними.

По-четверте, в основі всіх віртуальних моделей лежить їхній математичний опис. Цей математичний опис базується на фізичних основах, які повинні залишатися постійно в полі уваги дослідника від першого до останнього кроку моделювання.

Перераховані міркування викликають необхідність використання підходу, заснованого на математичному описі. При моделюванні складних об'єктів математичний опис може бути використаний при його дослідженні за «гладкою» складовою, тобто при моделюванні не враховуються імпульсні і високочастотні сигнали. При такому дослідженні використовуються безперервні параметри об'єкта. Цей підхід можна використати як перший етап у моделюванні. Результати, отримані на цьому етапі, можуть вважатися попередніми, з ними варто порівнювати результати, одержувані на більш складних віртуальних моделях.

Слід зазначити, що математичний опис окремих елементів дуже складних систем ефективно може бути заснований на використанні основного пакета *Matlab – Simulink*. При цьому окремі блоки представляються структурними схемами з головної бібліотеки *Simulink*. Тобто є можливість моделювання дуже складних систем з окремих, теж дуже складних блоків.

Контрольні питання до розділу 5

1. Охарактеризуйте метод моделювання як метод наукового пізнання.

2. Які існують особливості застосування методу математичного моделювання?

3. У чому полягає роль математичного моделювання у прикладних дослідженнях?

4. Назвіть основні особливості спостережень і вимірів при розробці моделі.

5. Розрізняють два типи невизначеності: «ширу» та «інформаційну». Охарактеризуйте їх.

6. Що таке поняття «адекватність моделі»? Наведіть приклади.

7. Що таке «верифікація» моделі? Наведіть приклади.

8. Наведіть загальну класифікацію математичних моделей.

9. Перерахуйте основні етапи математичного моделювання.

10. Який існує взаємозв'язок етапів моделювання?

11. Назвіть три складові процесу моделювання.

12. Що таке моделі детерміністського типу?

13. Чим відрізняються дескриптивні моделі від нормативних?

14. Чому застосовують дескриптивний підхід в імітаційному моделюванні?

Розділ 6

МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ ТА ЇХ ЕЛЕМЕНТІВ

6.1 Моделювання електричних машин

6.1.1 Загальні властивості стандартних моделей бібліотеки *Machines*

Блоки, призначені для моделювання різних електричних машин, знаходяться в бібліотеці *Machines* пакета *Sim Power Systems*. Для більшості типів машин існують два основні різновиди блоків: *pu units*, параметри в яких задаються у відносних одиницях (р.у. — *per unit*), і *SI units*, призначені для задання параметрів в абсолютних одиницях системи СІ. Обидва типи блоків побудовані на основі однієї і тієї ж математичної моделі, різниця полягає лише в способі задання параметрів. Розглянемо деякі загальні властивості, характерні для всіх стандартних моделей електричних машин бібліотеки *Machines*.

Кожна модель має два види портів — для зв'язку з блоками *Simulink* і *Sim Power Systems* (п. 2.1). *Simulink*-порти можуть бути або безрозмірними (якщо використовується модель *pu units*), або мати відповідну розмірність (якщо використовується модель *SI units*). *SPS*-порти є затискачами для включення моделі двигуна або генератора у віртуальне електричне коло.

Усі блоки електричних машин мають *Simulink*-вхід для подачі механічного впливу, за замовчуванням позначений T_m (для машин постійного струму — T_L) або P_m . Опція *Mechanical input* у вікні налаштування параметрів дає змогу вибрати з випадаючого списку блоки: *Torque* T_m — механічний момент навантаження на валу (або *Mechanical power* P_m — механічна потужність), *Speed* w — кутова частота обертання ротора. В залежності від цього змінюється позначення даного входу — T_m (P_m) або w . Якщо обрано опцію *Torque*,

то на цей вхід подається *Simulink*-сигнал, наприклад, від блока *Constant* або *Step*, що відповідає значенню навантажувального механічного моменту T_m або механічної потужності P_m , прикладених до машини. При виборі опції *Speed* w на вхід надходить сигнал частоти обертання w від зовнішнього пристрою, що відповідає генераторному режиму роботи, коли вал приводиться в обертання іншим двигуном. При цьому механічна частина моделі (момент інерції J або інерційна стала H) ігнорується.

Використання частоти обертання як механічного впливу дозволяє моделювати спільну роботу двох електричних машин за системою «двигун–генератор», а також досліджувати дану модель засобами пакетів *Sim Mechanics* і *Sim Driveline* середовища MATLAB. На рис. 6.1 зображена функціональна схема такої моделі. Дві електричні машини з'єднані через жорсткий вал, одна з них працює в режимі двигуна, друга — в режимі генератора. Сигнал частоти обертання w з виходу двигуна надходить на відповідний вхід генератора, тоді як з його виходу надходить електромагнітний момент T_e на вхід T_m двигуна. При цьому коефіцієнт K_w враховує одиниці виміру швидкостей обох машин (р.у. або рад/с) та їхнє передатне відношення. Коефіцієнт K_T враховує одиниці виміру моментів (р.у. або Нм), а також номінальні параметри машин. У налаштуваннях двигуна треба задати сумарне значення моменту інерції J_{12} , яке враховує моменти інерції обох машин (J_1 — двигуна, J_2 — генератора). Це значення обчислюється за формулою

$$J_{12} = J_1 + J_2 \left(\frac{w_2}{w_1} \right)^2.$$

У налаштуваннях генератора момент інерції ігнорується.

Вихід m усіх блоків призначений для виміру електричних і механічних сигналів (струмів, напруг, швидкості, моменту та ін.), набір і кількість яких залежить від типу конкретної машини. До даного виходу треба підключити блок *Bus Selector* (бібліотека *Simulink/Signal Routing*) і вибрати у його вікні налаштування необхідні сигнали. Виходи блока *Bus Selector* підключити

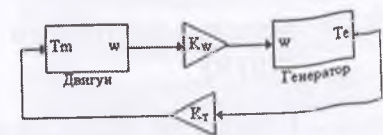


Рисунок 6.1 — Функціональна схема системи «двигун–генератор»

до осцилографа *Scope*. Також можна для цього використовувати блок *Machines Measurement Demux* (бібліотека *SPS / Machines*), у налаштуваннях якого потрібно вибрати тип машини й установити прапорці напроти вимірюваних сигналів.

У вікні налаштування кожного блока задаються параметри, характерні для даного типу електричної машини. Основні номінальні параметри: P_n – повна потужність, ВА; V_n – діюче значення лінійної напруги, В; f_n – частота, Гц (для машин змінного струму). Ці величини завжди задаються в одиницях системи СІ незалежно від типу блока. У версіях MATLAB, починаючи з 7.0.1 і вище, користувач має можливість вибрати основні параметри для конкретного типу машини зі списку, що випадає, у полі *Preset Model* вікна налаштування. При цьому у всіх інших полях параметри автоматично встановлюються в значення, що відповідають обраному варіанту.

Для машин змінного струму у вікні налаштування існує прапорець *Simulate saturation*, установка якого дозволяє моделювати нелінійну характеристику неробочого ходу. Якщо прапорець встановлено, то з'являється додаткове поле *Saturation parameters*, в яке вводяться значення струмів статора (для асинхронної машини) або струмів збудження (для синхронної машини) і відповідні їм значення вихідних напруг. Формат введення наступний: $[i_1, i_2, \dots (\text{p.u.}); v_1, v_2, \dots (\text{p.u.})]$. Кількість даних визначається обраною кількістю точок кривої. При цьому перша точка відповідає початку зони нелінійності (рис. 6.2). Якщо прапорець знятий, то залежність між струмом і напругою лінійна.

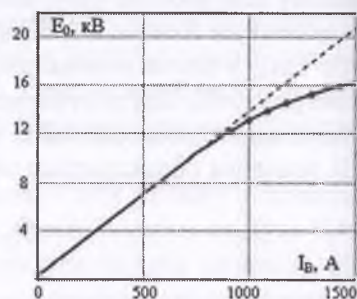


Рисунок 6.2 – Характеристика неробочого ходу синхронного генератора

6.1.2 Машини постійного струму

Стандартна модель машини постійного струму (МПС) у пакеті *SPS* представлена блоком (рис. 6.3), параметри якого задаються в одиницях системи СІ. Порти A+, A– призначені для підключення кола якоря (*armature*), порти F+, F– – для обмотки збудження (*field*).

Залежно від способу підключення цей блок дає змогу моделювати МПС з різними видами збудження: незалежним, послідовним, паралельним (рис. 6.4). Вхід T_L призначений для задання моменту навантаження. На виході m формується векторний сигнал, що складається з чотирьох елементів: кутової частоти обертання якоря ω_m (рад/с), струму якоря i_a (А), струму обмотки збудження i_f (А) та електромагнітного моменту T_e (Нм).

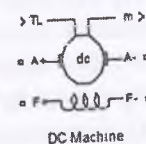


Рисунок 6.3 – Блок DC Machine

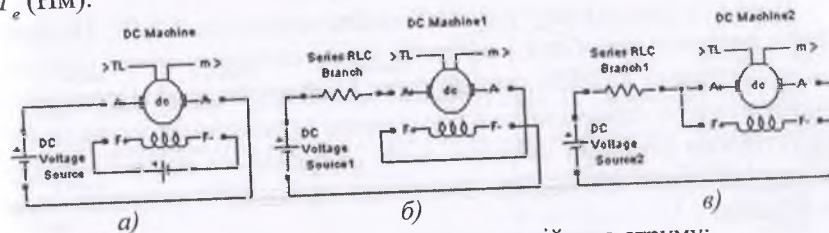


Рисунок 6.4 – Моделі двигуна постійного струму: з незалежним збудженням (а); послідовним збудженням (б); паралельним збудженням (в)

Вікно налаштування параметрів блока *DC Machine* зображено на рис. 6.5. У списку *Preset Model*, що випадає, наведені 23 різних типів МПС із зазначенням їхньої потужності, напруги якоря, частоти обертання і напруги обмотки збудження. Так, на рис. 6.5 як приклад обрана машина з порядковим номером 12, потужність якої 50 к.с. (*HP* – *horsepower*, 1HP = 746 ВА), напруга якоря 240 В, номінальна частота обертання 1750 об/хв (*RPM* – *revolutions per minute*), напруга обмотки збудження 300 В. Параметри в інших полях вікна налаштування відповідають даному типу машини.

Preset model	12: 50HP 240V 1750RPM Field 300V
Mechanical input	Torque TL
<input checked="" type="checkbox"/> Show detailed parameters	
Armature resistance and inductance [Ra (ohms) La (H)]	[0.4832 0.006763]
Field resistance and inductance [Rf (ohms) Lf (H)]	[84.91 13.39]
Field-armature mutual inductance Laf (H):	[0.7096]
Total inertia J (kg.m ²)	[0.2053]
Viscous friction coefficient Bm (N.m.s)	[0.007032]
Coulomb friction torque Tf (N.m)	[5.282]
Initial speed (rad/s):	[0]

Рисунок 6.5 – Вікно налаштування блока *DC Machine*

Прапорець *Show detailed parameters* дає змогу відобразити або приховати додаткові параметри:

R_a, L_a – опір та індуктивність якоря відповідно, Ом, Гн;

R_f, L_f – опір та індуктивність обмотки збудження, Ом, Гн;

L_{af} – взаємна індуктивність між обмоткою якоря й обмоткою збудження, Гн;

J – загальний момент інерції, кг·м²;

B_m – коефіцієнт в'язкого тертя, Нм·с;

T_f – момент сухого тертя, Нм;

Initial speed – початкова швидкість, рад/с.

Щоб відобразити внутрішню *Simulink*-модель блока *DC Machine*, треба виділити цей блок і вибрати в контекстному меню (при натисканні правої клавіші миші) пункт *Look under mask*. Електрична частина моделі представлена послідовно включеними активним та індуктивним опорами якірного кола і керованого джерела напруги, що моделює проти-ЕРС. Величина проти-ЕРС розраховується за виразом

$$E = k_E \cdot \omega,$$

де ω – частота обертання якоря; k_E – конструктивний коефіцієнт.

Електромагнітний момент T_e , що розвивається машиною, пропорційний струму якоря I_a :

$$T_e = k_T \cdot I_a,$$

де k_T – конструктивний коефіцієнт.

Режим роботи МПС залежить від знаків моменту навантаження T_L та електромагнітного моменту T_e :

$T_e T_L > 0$ – режим двигуна;

$T_e T_L < 0$ – режим генератора.

Механічна частина моделі складена з блоків *Simulink* відповідно до рівняння

$$J \frac{d\omega}{dt} = T_e - \text{sgn}(\omega) T_L - B_m \omega - T_f,$$

де $\text{sgn}(\omega) = 1$ при $\omega > 0$, $\text{sgn}(\omega) = -1$ при $\omega < 0$, $\text{sgn}(\omega) = 0$ при $\omega = 0$.

Варто пам'ятати, що якщо потрібно провести моделювання з ненульовими початковими умовами, то на вхід T_L повинно бути

подане значення моменту навантаження, що відповідає початковій швидкості.

На рис. 6.6 показана модель двигуна постійного струму з незалежним збудженням у режимі реостатного триступінчатого пуску. Пусковий реостат моделюється трьома резисторами, які у міру розгону двигуна шунтуються паралельними вимикачами в моменти часу t_1, t_2, t_3 відповідно (значення часу задані в налаштуваннях блоків *Breaker*). Момент навантаження залежить від частоти обертання й у даному випадку визначається співвідношенням $T_L = 0,2287 \omega_m$.

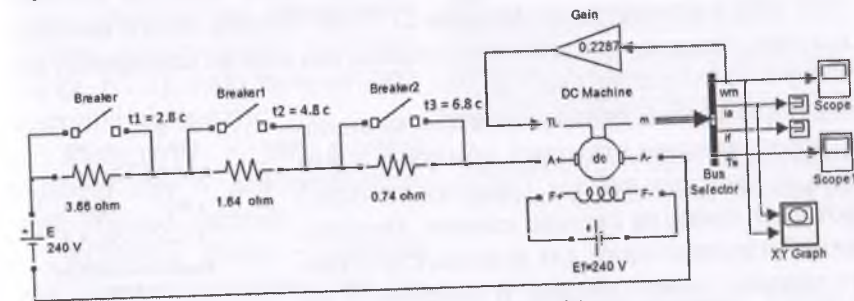


Рисунок 6.6 – Модель двигуна постійного струму в режимі реостатного пуску

Графіки перехідного процесу для частоти обертання ω і рушійного моменту M показані на рис. 6.7 та на рис. 6.8 зображена пускова діаграма двигуна, тобто залежність $\omega = f(M)$.

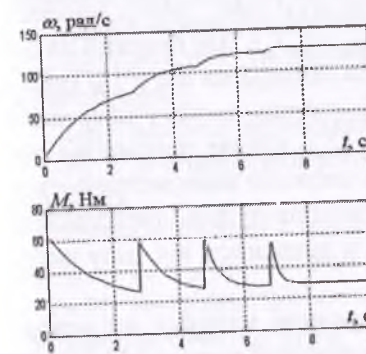


Рисунок 6.7 – Графіки перехідних процесів у двигуні постійного струму

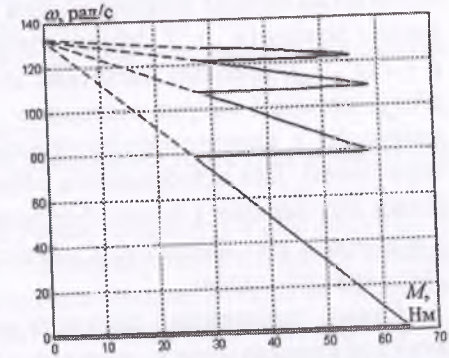


Рисунок 6.8 – Графік залежності $\omega = f(M)$

При цьому пунктирними лініями позначено сімйство статичних механічних характеристик, які відповідають додатковим опорам у колі якоря, а неперервними лініями показана динамічна механічна характеристика, отримана в процесі моделювання за допомогою блока *XY Graph*.

6.1.3 Асинхронні машини

Модель асинхронної машини (АМ) представлена в бібліотеці *Machines* двома блоками – *Asynchronous Machine pu Units* (рис. 6.9) і *Asynchronous Machine SI Units*. Обидва блоки використовують однакову математичну модель, яка описує асинхронну машину в *dq*-координатах.

Порти *A, B, C* на піктограмі блока відповідають виводам статорної обмотки, порти *a, b, c* – виводам ротора. Обмотки статора і ротора з'єднані за схемою «зірка». *Simulink*-вхід *Tm* призначений для задання статичного моменту навантаження. В залежності від використовуваної моделі машини (*pu* або *SI*) сигнал на цей вхід подається у відносних одиницях або в Нм відповідно. На виході *m* формується векторний сигнал, який містить 21 елемент: струми ротора (i_{ra}, i_{rb}, i_{rc}), їхні проекції на осі *q* і *d* (i_{rq}, i_{rd}), проекції магнітних потоків ротора на осі *q* і *d* ($\phi_{ir,q}, \phi_{ir,d}$), проекції напруг ротора (v_{rq}, v_{rd}), струми статора (i_{sa}, i_{sb}, i_{sc}), їхні проекції на осі *q* і *d* (i_{sq}, i_{sd}), проекції магнітних потоків статора на осі *q* і *d* ($\phi_{is,q}, \phi_{is,d}$), проекції напруг статора (v_{sq}, v_{sd}), кутову частоту обертання ротора (ω_m), електромагнітний момент (T_e), кутове положення ротора (θ). Для відображення обраних сигналів використовуються блоки *Bus Selector* і *Scope*. Результати вимірів будуть представлені у відносних або абсолютних одиницях, в залежності від типу використовуваної моделі.

Схеми заміщення і рівняння асинхронної машини, на основі яких побудована модель, наводяться в довідковій системі MATLAB. З *Simulink*-прототипом моделі АМ можна ознайомитися, відкривши файл *powerlib_models.mdl* (п. 2.1). На рис. 6.10 показане вікно налаштування блока *Asynchronous Machine pu Units*. Верхнє поле

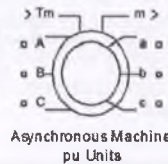


Рисунок 6.9 – Блок *Asynchronous Machine pu Units*

Preset Model, як і в моделі МПС (п. 6.1.2), призначене для вибору одного з пропонуванних варіантів асинхронних машин із заданими номінальними параметрами: потужність, діюче значення лінійної напруги статора, частота струму, кутова частота обертання ротора. Поле *Mechanical input* визначає вид механічного впливу, прикладеного до машини (п. 6.1.1). Інші поля містять наступні параметри:

Rotor type – тип ротора: *Squirrel-cage* – короткозамкнений, *Wound* – фазний;

Reference frame – система відліку, застосовувана для перетворення *ABC*-координат у *dq*-координати і навпаки. Вибирається один із трьох варіантів: система нерухома відносно статора (*Stationary*), система нерухома відносно ротора (*Rotor*), система, що обертається разом з полем (*Synchronous*);

P_n, V_n, f_n – повна потужність (ВА), діюче значення лінійної напруги статора (В), частота струму (Гц) відповідно;

R_s, L_{ls} – опір та індуктивність статора, р.у.;

R_r, L_{lr} – опір та індуктивність ротора, р.у. (зведені до параметрів статора);

L_m – взаємна індуктивність, р.у.

H, F, p – інерційна стала (с), коефіцієнт тертя (р.у.), число пар полюсів;

Initial conditions – початкові умови. Задані у вигляді рядка [$s, \theta, I_A, I_B, I_C, \phi_A, \phi_B, \phi_C$], що визначає стан асинхронної машини в момент початку моделювання: s – початкове ковзання; θ – кутове положення ротора, град; I_A, I_B, I_C – амплітуди струмів статора, р.у.; ϕ_A, ϕ_B, ϕ_C – фазові кути струмів статора, град.

Ці дані можуть бути розраховані за допомогою блока *Powergui* (п. 6.2.3). Рядок [1, 0, 0, 0, 0, 0, 0, 0] відповідає нульовим початковим умовам ($s = 1$ при нульовій частоті обертання).

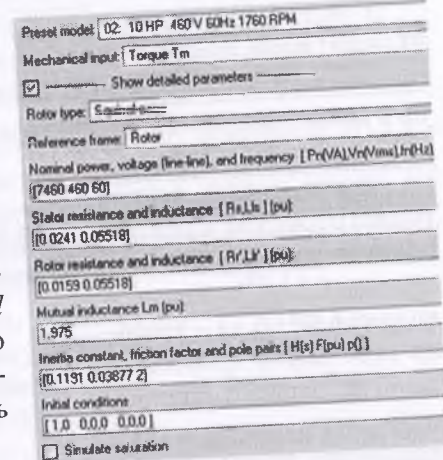


Рисунок 6.10 – Вікно налаштування блока *Asynchronous Machine pu Units*

У моделі АМ з параметрами в системі CI (*Asynchronous Machine SI Units*) вікно налаштування виглядає аналогічно, з тією лише різницею, що замість відносних одиниць використовуються абсолютні, а замість інерційної сталої H задається момент інерції J , кг·м².

На рис. 6.11 зображена проста модель асинхронного двигуна в режимі прямого пуску з навантаженням $0,2M_{\text{ном}}$ на валу, а на рис. 6.12 і 6.13 – графіки перехідних процесів у даній моделі і динамічні механічні характеристики відповідно. Між трифазним джерелом і двигуном включене додаткове малопотужне активне навантаження, яке потрібно для того, щоб уникнути повідомлення про помилку при запуску моделювання. Помилка виникає при послідовному з'єднанні джерела струму, на основі якого побудована модель двигуна, та індуктивності, що входить до складу трифазного джерела напруги (див. п. 2.8). Інший спосіб уникнути повідомлення про помилку – задати індуктивність джерела рівною нулю.

На виході двигуна вимірюються кутова частота обертання ω і електромагнітний момент M , у відносних одиницях. Для того, щоб одержати обидва сигнали в одному вікні (див. рис. 6.12), необхідно в налаштуваннях блока *Bus Selector* установити прапорець *Output as a bus*. При цьому усі вихідні сигнали об'єднуються в одну спільну шину.

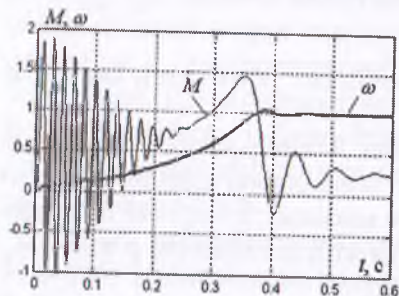


Рисунок 6.12 – Графіки перехідних процесів в асинхронному двигуні

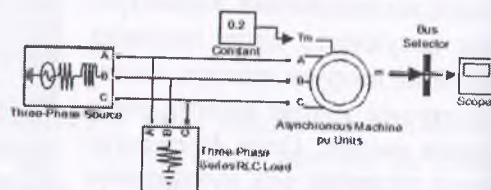


Рисунок 6.11 – Модель асинхронного двигуна для режиму прямого пуску

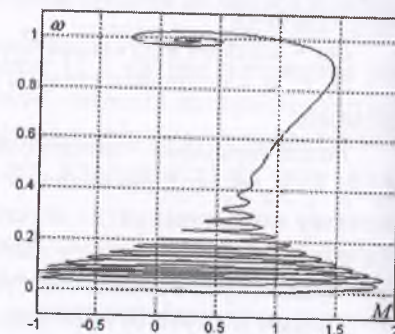


Рисунок 6.13 – Графік залежності $\omega = f(M)$

6.1.4 Синхронні машини

6.1.4.1 Спрощена модель синхронної машини

Спрощена модель трифазної синхронної машини (СМ) з неявнополюсним ротором представлена блоком *Simplified Synchronous Machine* (рис. 6.14). Порти моделі A, B, C є виводами статорної обмотки. Три фази з'єднані за схемою «зірка». Електрична частина кожної фази моделі складається з джерела напруги, включеного послідовно з активно-індуктивним опором, який являє собою внутрішній опір машини. При цьому активний опір може бути нульовим, але індуктивність повинна мати ненульове значення. У даній моделі не враховується індуктивність розсіювання, само- і взаємодія якоря, обмотки збудження і демпферної обмотки. Ефект демпферної обмотки приблизно враховується коефіцієнтом демпфірування K_d . Модель механічної частини блока відповідає рівнянню

$$\omega(t) = \omega_0 + \Delta\omega(t), \quad (6.1)$$

де $\omega(t)$ – поточне значення кутової частоти обертання ротора; $\omega_0 = 1$ р.у. – синхронна частота обертання; $\Delta\omega(t)$ – відхилення частоти обертання від синхронної:

$$\Delta\omega(t) = \frac{1}{2H} \int_0^t (T_m - T_e) dt - K_d \Delta\omega(t);$$

H – інерційна стала; T_m, T_e – механічний та електромагнітний моменти відповідно; K_d – коефіцієнт демпфірування.

Рівнянню (6.1) відповідає структурна схема на рис. 6.15.

Розглянемо призначення входів і виходів блока *Simplified Synchronous Machine*.

P_m – вхід для задання механічної потужності на валу машини. Це може бути постійний

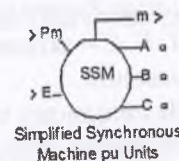


Рисунок 6.14 – Блок *Simplified Synchronous Machine*

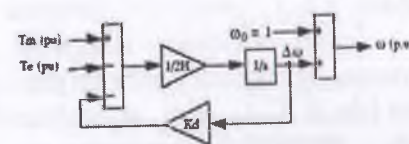


Рисунок 6.15 – Модель механічної частини блока *Simplified Synchronous Machine*

вплив або сигнал з виходу керуючого блока *Steam Turbine and Governor* (п. 6.2.1). Режим роботи СМ визначається знаком сигналу на вході Pm :

$Pm > 0$ – режим генератора;

$Pm < 0$ – режим двигуна;

E – діюче значення лінійної напруги внутрішнього джерела. Це може бути постійний вплив або сигнал з виходу блока збудження *Excitation System* (п. 4.1.4.3);

A, B, C – затискачі трьох фаз статора, до яких підключається навантаження.

На вихідному порту m формується векторний сигнал, що містить 12 елементів: струми (i_{sa}, i_{sb}, i_{sc}), напруги (v_{sa}, v_{sb}, v_{sc}) та ЕРС (e_a, e_b, e_c) обмотки статора, кутове положення (θ_m) і частоту обертання ротора (ω_m), а також електричну потужність машини (P_e).

Сигнали на *Simulink*-входах Pm, E і виході m можуть бути у відносних або абсолютних одиницях, у залежності від типу використовуваної моделі.

Вікно налаштування параметрів для блока *Simplified Synchronous Machine pu Units* зображене на рис. 6.16. Розглянемо призначення полей.

Connection type – визначає схему з'єднання обмоток генератора: «зірка» без нульового проводу (3-wire Y) або «зірка» з нульовим проводом (4-wire Y). Якщо обрано чотирипровідну схему, то на піктограмі блока з'являється додатковий вхід N для підключення нульового проводу.

H, Kd, p – інерційна стала (с), коефіцієнт демпфування і число пар полюсів відповідно. Для моделі *SI Units* замість параметра H заданий момент інерції J , кг·м².

R, X – внутрішній активний та індуктивний опір кожної фази, р.у.

Initial conditions – початкові умови. Задаються у вигляді рядка $[dw, \theta, I_a, I_b, I_c, \phi_a, \phi_b, \phi_c]$, де dw – початкове відхилення кутової частоти обертання ротора (у % від номінальної), а інші змінні аналогічні до наведених

Connection type:	3-wire Y
Nom. power, L-L volt., and freq [Pn(WA) Vn(Vms) fn(Hz)]	[107e6 13800 60]
Inertia, damping factor and pairs of poles [H(sec) Kd(pu) Tp(pu_w) p()	[3.7 0.20]
Internal impedance [R(pu) X(pu)]	[0.02 0.3]
Init. cond. [dw(%) th(deg) ia,ib,ic(pu) pha,phb,phc(deg)]	[0.0 0.0 0.0 0.0 0.0]

Рисунок 6.16 – Вікно налаштування блока *Simplified Synchronous Machine pu Units*

у п. 6.1.3 для асинхронної машини. За замовчуванням задане $dw = 0$, що відповідає обертанню ротора з номінальною швидкістю в початковий момент часу. Якщо потрібно моделювати процес з нульовою початковою швидкістю, то для параметра dw треба задати значення $-99,9$ (при значенні $dw = -100$ видається повідомлення про помилку).

У моделі синхронної машини з параметрами в системі *SI* (*Simplified Synchronous Machine SI Units*) вікно налаштування виглядає аналогічно, але замість відносних одиниць (р.у.) використовуються абсолютні.

6.1.4.2 Повна модель синхронної машини

Крім спрощеної моделі, у пакеті *Sim Power Systems* існують три блоки для детального моделювання синхронної машини – *Synchronous Machine Fundamental* (у *pu*- і *SI*-параметрах) і *Synchronous Machine pu Standard*. Усі вони мають однакові піктограми (рис. 6.17) і використовують одну і ту ж математичну модель, розходження полягає тільки в способі задання параметрів.

Модель механічної частини даних блоків така ж, як і в спрощеному блоці *Simplified Synchronous Machine* (п. 6.1.4.1), модель електричної частини являє собою систему 6-го порядку у формі простору станів і враховує динаміку статора, обмотки збудження і демпферної обмотки. Математичний опис базується на dq -системі координат, нерухомій відносно ротора. Рівняння синхронної машини і її схема заміщення наводяться в довідковій системі *MATLAB*.

Залежно від знаку механічної потужності, що подається на вхід Pm , кожен блок може працювати як у режимі генератора ($Pm > 0$), так і в режимі двигуна ($Pm < 0$). На вхід Vf (*field voltage*) подається напруга збудження від регулятора *Excitation System* (п. 6.1.4.3) або від блока *Constant*. На виході m формується векторний сигнал, що містить

22 елементи: струми статора (i_{sa}, i_{sb}, i_{sc}), їхні проекції на осі q і d (i_q, i_d), струм збудження (i_{fa}), проекції струмів демпферної обмотки на осі q і d (i_{ka1}, i_{ka2}, i_{kd}), проекції намагнічуючого потоку (ϕ_{imq}, ϕ_{imd}), проекції напруг статора (v_{sq}, v_{sd}), відхилення кута ротора (d_theta), кутову

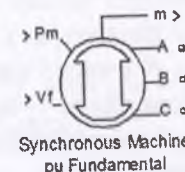


Рисунок 6.17 – Блок *Synchronous Machine Fundamental*

частоту обертання ротора (ω_m), повну електричну потужність (P_e), відхилення швидкості ротора (dw), механічний кут ротора (θ), електромагнітний момент (T_e), кут навантаження (δ), активну і реактивну потужності на виході машини (P_{eo} , Q_{eo}).

У полі *Rotor type* вікна налаштування всіх трьох блоків вибирається тип ротора: *Salient-pole* – явнополюсний, *Round* – неявнополюсний. Для блока *Synchronous Machine SI Fundamental* (рис. 6.18) у полі з номінальними параметрами, крім потужності, напруги і частоти, задається також номінальне значення струму збудження i_{fd} . Воно застосовується для розрахунку номінальної напруги збудження V_{fd} за якої напруга V_t (terminal voltage) на виході генератора буде номінальною. Щоб одержати значення V_{fd} треба встановити прапорець *Display Vfd which produces a nominal Vt* у нижній частині вікна налаштування. При цьому з'явиться вікно з розрахованим V_{fd} . Отримане значення може бути подане на вхід V_f за допомогою блока *Constant*.

В інших полях вікна налаштування блока *Synchronous Machine Fundamental* (як *pu*-, так і *SI*-) задаються опори й індуктивності обмотки статора (*Stator: Rs, Ll, Lmd, Lmq*), обмотки збудження (*Field: Rf, Lfd*) і демпферної обмотки (*Dampers: Rkd', Lkd', Rkq', Lkq'*). Індекси параметрів мають наступні значення: d, q – проекції змінних на осі d і q ; l, m – індуктивності розсіювання (*leakage*) і намагнічування (*magnetizing*); f, k – змінні кола збудження і демпферної обмотки. Усі параметри зведені до статора. Задаються також інерційна стала H (момент інерції J для моделі в *SI*-параметрах), коефіцієнт тертя F і число пар полюсів p . Момент тертя T_f пропорційний до кутової швидкості обертання ротора: $T_f = F \cdot \omega$.

Рядок задання початкових умов *Initial conditions* у цілому такий же, як і для спрощеної

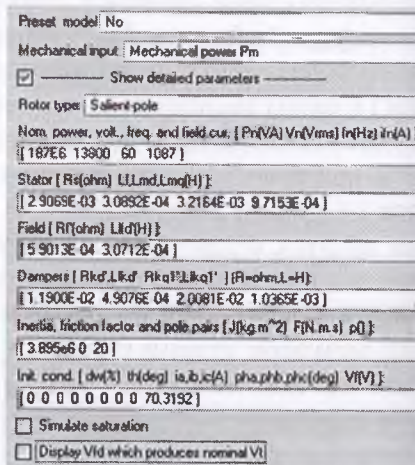


Рисунок 6.18 – Вікно налаштування блока *Synchronous Machine SI Fundamental*

синхронної машини (п. 6.1.4.1), з тією відмінністю, що додається параметр V_f – початкове значення напруги збудження. Якщо моделювання виконується при номінальному початковому збудженні, то для *pu*-блока $V_f = 1$ р.у., а для *SI*-блока як V_f треба ввести значення V_{fd} розраховане при установці відповідного прапорця (див. рис. 6.18).

У вікні налаштування блока *Synchronous Machine pu Standard* замість параметрів статорної, роторної і демпферної обмотки задаються реактивні опори (*reactances*) по осях d і q : X_d, X_q – синхронні, X_d', X_q' – перехідні, X_d'', X_q'' – надперехідні, X_l – опір розсіювання. Також задаються значення сталих часу за осями d і q для обраного режиму – неробочого ходу (*open-circuit*) чи короткого замикання (*short-circuit*).

Модель електроенергетичної системи з використанням блока *Synchronous Machine pu Fundamental* наведена на рис. 6.27.

6.1.4.3 Система збудження і регулятор напруги СМ

Блок *Excitation System* (рис. 6.19) являє собою модель системи збудження і регулятора напруги синхронної машини. Внутрішня модель блока, що складена з *Simulink*-елементів, показана на рис. 6.20. Збудник (*Exiter*) і регулятор (*Main regulator*) моделюються аперіодичними ланками першого порядку з параметрами K_e , T_i і K_a , T_a відповідно. У моделі також присутні фільтр нижніх частот (*Low-pass filter*), стабілізуючий компенсатор (*Lead-lag compensator*) і демпфуюча ланка (*Damping*). Параметри всіх цих елементів задаються у вікні налаштування блока *Excitation System*.

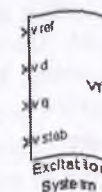


Рисунок 6.19 – Блок *Excitation System*

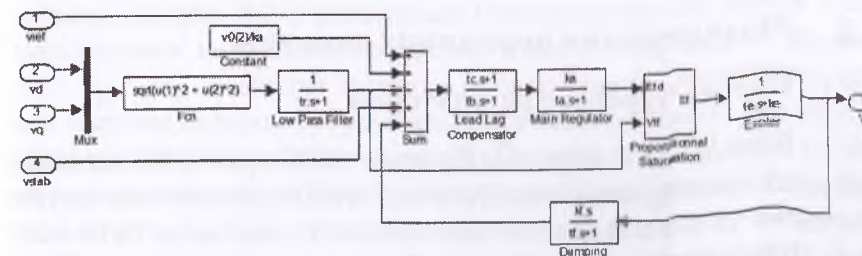


Рисунок 6.20 – Внутрішня модель блока *Excitation System*

Блок *Proportional saturation* (рис. 6.20) – це підсистема, яка реалізує обмеження вихідної напруги регулятора. Верхня $E_{f_{\max}}$ і нижня $E_{f_{\min}}$ границі обмеження, а також масштабуючий коефіцієнт K_p задаються у вікні налаштування. Якщо $K_p = 0$, то верхня границя дорівнює значенню $E_{f_{\max}}$, а якщо $K_p > 0$, то верхня границя обмеження дорівнює $K_p \cdot V_{\varphi}$, де V_{φ} – напруга генератора на виході фільтра нижніх частот.

Крім розглянутих вище параметрів, у вікні налаштування задаються також початкові умови моделювання: V_{α} , V_{β} – початкові значення напруги на затискачах генератора і напруги на обмотці збудження відповідно, р.у. Значення V_{α} звичайно дорівнює 1 р.у. Значення V_{β} розраховується за допомогою блока *Powergui* (п. 6.2.3).

Розглянемо призначення входів і виходу блока *Excitation System*. Сам блок фактично є підсистемою, складеною з елементів *Simulink*, тому він не призначений для включення у віртуальне електричне коло. Отже, усі його входи і вихід є *Simulink*-портами:

v_{ref} – необхідне значення напруги на затискачах статора, р.у. Сигнал надходить від блока *Constant*. Якщо потрібна номінальна напруга, то значення константи дорівнює 1;

vd , vq – поточні значення проєкцій напруги статора на осі d і q , р.у. Сигнали надходять з вимірювального виходу m блока *Synchronous Machine* (через блок *Selector*);

v_{stab} – додаткова стабілізація коливань напруги. До даного входу може підключатися блок *Power System Stabilizer*, або, якщо стабілізація не потрібна, вхід можна залишити вільним;

V_f – вихід, з якого напруга збудження подається на однойменний вхід блока *Synchronous Machine*.

Застосування блока *Excitation System* показано на прикладі рис. 6.27.

6.2 Моделювання первинних двигунів

6.2.1 Парова турбіна з регулятором

Блок *Steam Turbine and Governor (STG)*, рис. 6.21, містить у собі модель чотириступінчатої парової турбіни, регулятора частоти обертання та багатомасового вала (кількість мас може бути від 1 до 4). При необхідності можна відобразити внутрішню модель блока (рис. 6.22), якщо вибрати в контекстному меню (при натисканні

правої клавіші миші) пункт *Look under mask*. Аналогічно відображаються моделі всіх підсистем (рис. 6.22).

Регулятор частоти обертання (блок *Speed regulator* на рис. 6.22, його внутрішня модель – на рис. 6.23) складається з пропорційної ланки з коефіцієнтом K_p , реле частоти обертання *Speed Relay* і сервомотора, що керує відкриванням клапана. Коефіцієнти K_p , R_p , сталі часу реле T_{sr} і сервомотора T_{sm} , а також ширина зони нечутливості (*dead zone*) Dz задаються у вікні налаштування блока *STG*.

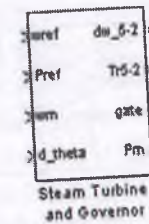


Рисунок 6.21 – Блок *STG*

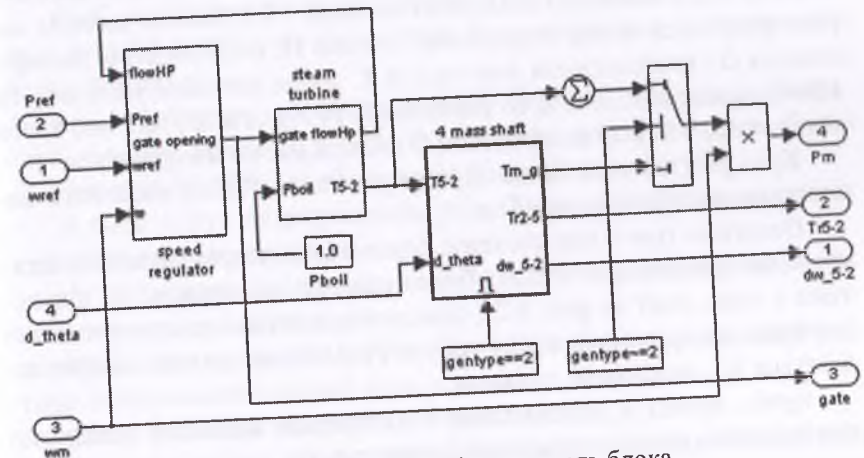


Рисунок 6.22 – Внутрішня модель блока *Steam Turbine and Governor*

Парова турбіна (блок *Steam turbine* на рис. 6.22, його внутрішня модель – на рис. 6.24) складається з чотирьох частин, кожна з яких моделюється аперіодичною ланкою першого порядку зі сталими часу T_2 , T_3 , T_4 , T_5 . Перша частина – це парозбірник (*steam chest*), інші три частини являють собою пароперегрівник (*reheater*) або пропускний трубопровід (*crossover piping*). Паровий котел у даній системі не моделюється, його тиск прийнятий постійним, рівним 1 р.у. Коефіцієнти F_2 , F_3 , F_4 , F_5 використовуються для розподілу потужності турбіни по відповідних частинах вала. Значення коефіцієнтів F_2 – F_5 , а також сталих часу T_2 – T_5 задаються у вікні налаштування.

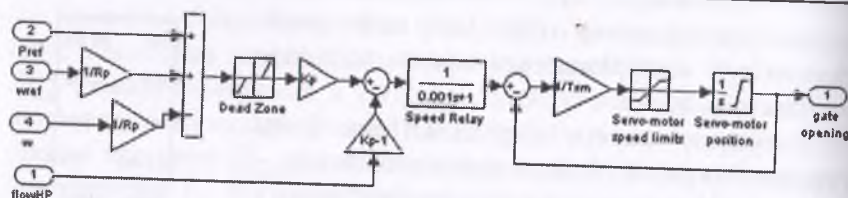


Рисунок 6.23 – Внутрішня модель підсистеми *Speed regulator*

Чотиримасовий вал (блок 4 *mass shaft* на рис. 6.22) разом із синхронною машиною, яка підключається до блока *STG*, утворює систему, що складається з п'яти мас. При цьому маса машини вважається першою і позначається #1. Найближча до машини частина вала позначена #2, а найбільш віддалена частина – #5. Кожна частина характеризується своєю інерційною сталою H , коефіцієнтом демпфіювання D і коефіцієнтом жорсткості K . Якщо потрібно моделювати кількість мас менше 4-х, то значення H та F для відсутніх мас треба установити в 0. Їхні коефіцієнти D і K при цьому ігноруються.

Крім розглянутих вище параметрів блока *STG*, у вікні його налаштування також задаються:

Generator type – тип системи: одномасова (*single-mass*) або багатомасова (*multimass*). Якщо обрано одну масову модель, то підсистема 4 *mass shaft* на рис. 6.22 цілком ігнорується програмою, а всі значення моментів з виходу турбіни T5-2 підсумовуються і прикладаються до синхронної машини;

vgmin, *vgmax* – максимальне і мінімальне значення швидкості переміщення клапана для подачі пари, р.у./с;

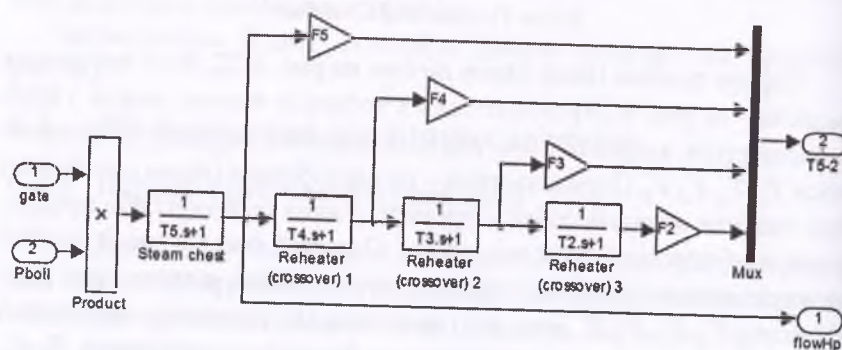


Рисунок 6.24 – Внутрішня модель підсистеми *Steam turbine*

gmin, *gmax* – максимальна і мінімальна границі регулювання клапана, р.у. (1 – повністю відкритий, 0 – повністю закритий);

Nominal speed – номінальна частота обертання синхронного генератора, який приводиться в рух паровою турбіною, рад/с;

Pm0, *th0* – початкові значення механічної потужності (р.у.) і кута повороту вала θ (град.) відповідно. Для одномасової моделі потрібно задати тільки потужність. Ці значення автоматично розраховуються за допомогою блока *Powergui* (п. 6.2.3).

Входи і виходи блока *Steam Turbine and Governor*:

wref – необхідне значення частоти обертання, р.у. Зазвичай на цей вхід подається сигнал від блока *Constant*, рівний 1;

Pref – необхідне значення електричної потужності, р.у. Задається у вигляді константи, рівної *Pm0*. Розраховується за допомогою блока *Powergui*;

wm – фактичне значення кутової частоти обертання генератора, р.у. На даний вхід подається сигнал *wm* з виходу *m* блока *Synchronous Machine* (за допомогою блока *Selector*);

d_theta – кут навантаження, рад. Також надходить з виходу синхронного генератора;

dw_5-2 – вектор відхилень частоти обертання для кожної частини багатомасової моделі вала, р.у.;

Tr5-2 – вектор, який містить значення моментів для кожної частини багатомасової моделі вала, р.у.;

gate – величина відкриття клапана турбіни, р.у.;

Pm – механічна потужність, що подається на відповідний вхід блока *Synchronous Machine*, р.у.

Модель, до складу якої входить блок *Steam Turbine and Governor*, зображена на рис. 6.27.

6.2.2 Дизель

У бібліотеці стандартних елементів модель дизеля відсутня. Однак її можна скласти самостійно з *Simulink*-блоків, або взяти готову з демонстраційного файлу *power_machines.mdl* пакета *Sim Power Systems* (у більш ранніх версіях MATLAB файл називається *psbmachines.mdl*). Щоб відкрити файл, треба ввести його ім'я (без розширення *.mdl*) після курсору в командному вікні MATLAB і натиснути клавішу *Enter*. Модель дизеля з регуляторами має назву

Diesel Engine Speed and Voltage Control (рис. 6.25). Вона виконана у вигляді підсистеми, що складається з дизеля з регулятором частоти обертання *Governor and Diesel Engine* і блока *Excitation System* (п. 6.1.4.3). На входи підсистеми подаються бажані значення частоти обертання генератора w_{ref} та його напруги V_{tref} (у вигляді константи, рівної 1 р.у.), а також вектор вимірюваних змінних безпосередньо з виходу m блока *Synchronous Machine*. З виходів знімаються механічна потужність P_m , напруга збудження V_f , необхідна напруга на затискачах генератора V_t і його частота обертання w . Значення P_m і V_t надходять на однойменні входи синхронної машини.

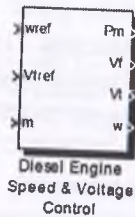


Рисунок 6.25 – Блок *Diesel Engine Speed and Voltage Control*

Внутрішня модель блока *Governor and Diesel Engine* представлена на рис. 6.26. На входи подаються бажана w_{ref} і дійсна w частоти обертання, з виходу знімається механічна потужність P_m .

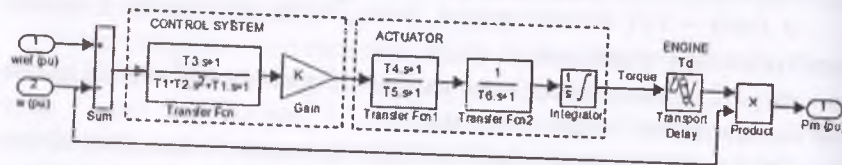


Рисунок 6.26 – Внутрішня модель блока *Governor and Diesel Engine*

Усі сигнали у відносних одиницях. Модель виконана у вигляді структурної схеми, що складається з регулятора частоти обертання (*Control system*), виконавчого механізму (*Actuator*) і двигуна (*Engine*). У вікні налаштування блока задаються параметри всіх передатних функцій, що входять до складу моделі, а також T_{max} , T_{min} – верхня і нижня границі обмеження моменту, P_{m0} – початкове значення механічної потужності на виході.

6.2.3 Ініціалізація моделей електричних машин

При моделюванні електроенергетичної системи, яка містить стандартні моделі бібліотеки *Machines*, потрібно знати значення постійних величин, що подаються на входи деяких блоків. Як правило, це значення механічної потужності і напруги збудження

синхронного генератора. Вони можуть бути розраховані й автоматично заведені в налаштування блоків *Constant* за допомогою спеціального інтерфейсу *Load Flow and Machine Initialization* блока *Powergui*.

Крім того, у моделі, до складу якої входять одна або декілька машин змінного струму, цей інтерфейс може бути використаний для розрахунку початкових умов у тих випадках, коли вони мають бути ненульовими. Іншими словами, передбачається, що в момент початку моделювання машини працюють у сталому режимі, тобто їхній пуск був здійснений заздалегідь. Процес розрахунку й установки ненульових початкових умов називається ініціалізацією. При цьому програмою автоматично розраховуються початкові значення струмів і фазових кутів всіх електричних машин змінного струму, а також початкова потужність P_{m0} блоку *Steam Turbine and Governor* і напруга збудження V_f блока *Excitation System*.

Розглянемо порядок ініціалізації на прикладі моделі на рис. 6.27. До шин генераторної установки, яка складається з синхронної машини (187 кВА, 460 В), парової турбіни і системи збудження, підключено асинхронний двигун (120 кВА) і активно-ємнісне навантаження (20 кВт, 10 кВАр).

Значення всіх сигналів у блоках *Constant* даної моделі спочатку були встановлені в 1. Для виконання ініціалізації треба відкрити блок *Powergui* і вибрати пункт *Load Flow and Machine Initialization*.

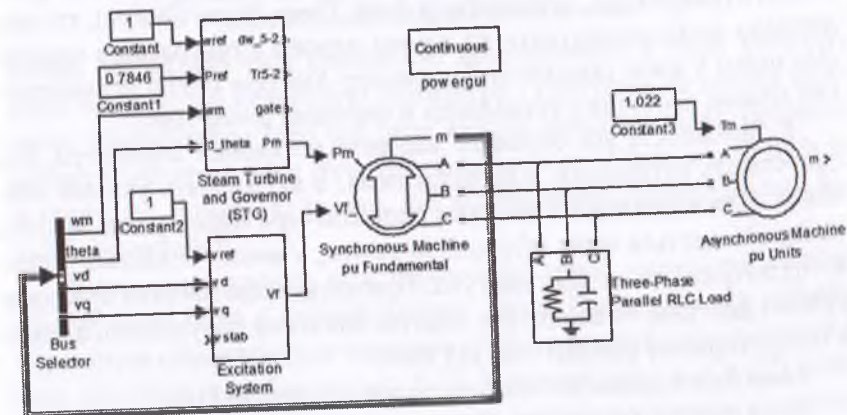


Рисунок 6.27 – Модель електроенергетичної системи

При цьому відкривається вікно, аналогічне зображеному на рис. 6.28. У його лівій частині *Machines load flow* відображаються значення змінних для машин, які присутні у моделі: амплітуди напруг і струмів, їхні початкові фази, активна і реактивна потужності, момент ковзання (для асинхронної машини) і напруга збудження (для синхронної машини). У правій частині вікна наведені параметри, які потрібно налаштувати перед виконанням ініціалізації. Розглянемо призначення основних полів налаштування.

Machines – поле відображає імена блоків синхронних і асинхронних машин, які присутні у моделі. Для налаштування параметрів ініціалізації конкретної машини треба вибрати її ім'я в списку.

Bus type – вказується, в якій формі задаються параметри для синхронної машини. Можливі три варіанти:

- *P&V generator* – задаються бажана лінійна напруга на затискачах машини (*Terminal voltage* U_{AB}), В, та її активна потужність (*Active power*), Вт;
- *P&Q generator* – задаються активна і реактивна потужності (*Reactive power*), Вт, Вар;
- *Swing Bus* – задаються бажана напруга, В, очікувана активна потужність (*Active power guess*), Вт, та початковий фазовий кут напруги U_{AN} фази А (*Phase of U_{AN} voltage*), град. Очікувана потужність – це сума активних потужностей усіх споживачів.

Якщо в моделі немає джерела, в якому задається початкова фаза напруги (наприклад, стандартний блок *Three-Phase Source*), то таке джерело треба реалізувати на основі одного з генераторів моделі. Для цього у вікні ініціалізації параметр *Bus type* однієї із синхронних машин необхідно установити в значення *Swing Bus*.

У залежності від обраного варіанта установки параметра *Bus type* стають активними відповідні поля, в які вводять указані значення. Для асинхронної машини поле *Bus type* недоступне, замість нього з'являється поле *Mechanical power*, у якому задається механічна потужність на валу двигуна. Причому, якщо машина працює в режимі двигуна, то вводиться додатне значення потужності, а якщо в генераторному режимі – то від'ємне.

Load flow frequency – частота змінної напруги, Гц.

Load flow initial condition – початкові умови. За замовчуванням цей параметр встановлено у значення *Auto*, при цьому початкові

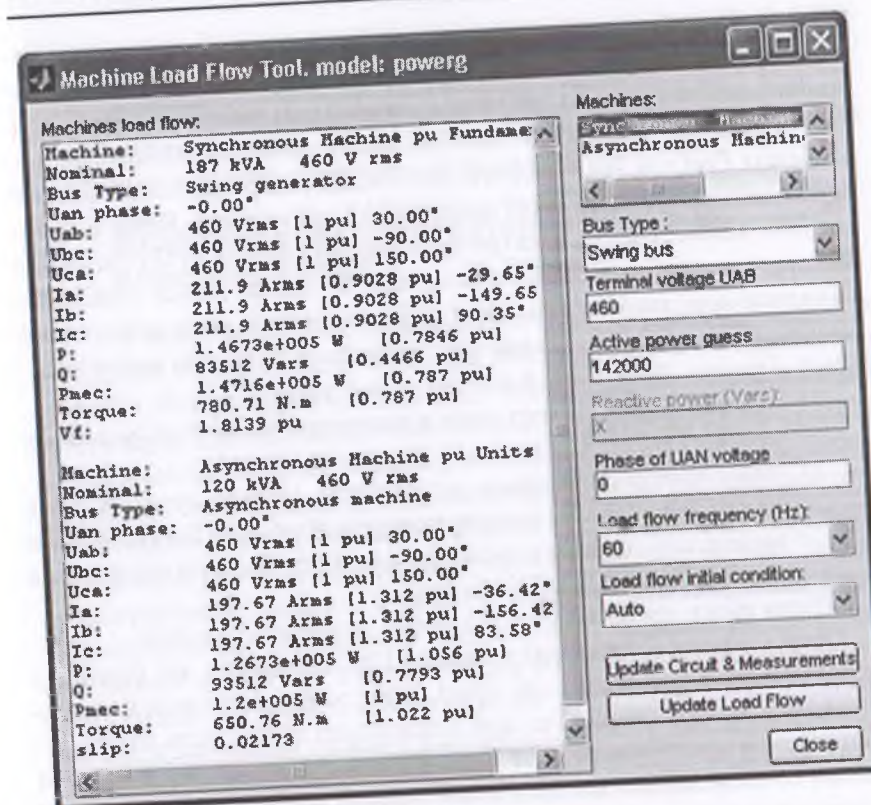


Рисунок 6.28 – Вікно налаштування інтерфейсу
Load Flow and Machine Initialization

умови розраховуються програмою автоматично. У тому випадку, якщо були задані некоректні параметри і програма не змогла здійснити ініціалізацію (про що буде видане повідомлення), потрібно вибрати значення *Start from previous solution*. Після цього будуть встановлені початкові умови, які відповідають попередньому запуску інтерфейсу.

Після встановлення всіх параметрів треба натиснути кнопку *Update Load Flow* у нижній частині вікна. Відповідно до заданих параметрів обновляються значення струмів, напруг і потужностей у полі *Machines load flow* на рис. 6.28. У моделі (див. рис. 6.27) автоматично змінюються значення сталих величин у блоках *Constant*, підключених до входів *Pref* турбіни і *Tm* двигуна (за замовчуванням

рівні 1). Постійна величина на вході T_m двигуна (момент навантаження у відносних одиницях) приблизно дорівнює 1, оскільки при ініціалізації потужність на валу двигуна була задана номінальною (120 кВА), що відповідає номінальному моменту навантаження. Значення $P_{ref} = 0,7846$ свідчить про те, що генератор завантажений приблизно на 78 % від своєї номінальної потужності. Якщо відкрити вікна налаштування електричних машин і блоків *STG* і *Excitation System*, то можна перекоонатися, що в полях *Initial conditions* з'явилися нові значення, які відповідають ненульовим початковим умовам. Далі треба додатково настроїти блок *STG*. Для цього необхідно відкрити його вікно й виконати наступні дії:

- 1) установити параметр g_{min} в значення, рівне P_{ref} (у даному випадку 0,7846), а g_{max} – у значення, рівне 1;
- 2) якщо використовується одномасова модель турбіни, то в рядку *Initial power* ввести значення P_{ref} . Для багатомасової моделі цей параметр змінювати не потрібно (він змінюється автоматично при ініціалізації).

Після цього треба запустити процес розрахунку моделі.

Якщо в моделі присутні вимикачі (блок *Breaker*), які спочатку перебувають в розімкнутому стані, а в процесі моделювання підключають навантаження, то при ініціалізації потужність цього навантаження враховуватися не буде. Іншими словами, всі початкові умови обчислюються для того стану моделі, в якому вона перебуває в момент початку моделювання. Наприклад, якщо на початку процесу генератор знаходиться в неробочому ході, а через деякий час до нього підключається навантаження, то початкові умови будуть розраховані саме для неробочого ходу. У такому випадку для адекватної установки вихідних даних, що відповідають роботі генератора під навантаженням, треба установити вимикач у замкнутий стан і виконати ініціалізацію, а вже після цього розімкнути його.

У тому випадку, коли одна з машин у момент початку моделювання повинна працювати в усталеному режимі, а інша машина запускається через деякий час після початку, тоді після ініціалізації всієї моделі треба обнулити початкові умови другої машини. Наприклад, до шин генератора, що працює, у визначений момент часу підключається асинхронний двигун за допомогою блока *Breaker*. При цьому початкові умови генератора мають бути ненульові, а для двигуна – нульові, оскільки його пуск здійснюється вже в процесі

моделювання. У цьому випадку треба виконати ініціалізацію, потім відкрити вікно налаштування асинхронної машини й у поле *Initial condition* ввести рядок [1,0,0,0,0,0,0].

Процес ініціалізації необхідно щоразу проводити заново після внесення змін у параметри елементів моделі.

6.3 Моделювання систем автоматизованого електропривода

У версії MATLAB 2008b у бібліотеці стандартних елементів пакета *Sim Power Systems* появився додатковий розділ *Application Libraries /Electric Drives Library*, у якому зібрано готові моделі електроприводів постійного і змінного струмів. Кожен блок містить у собі модель силового перетворювача, електродвигуна і системи керування. Завдяки такому нововведенню істотно спростилося розробка моделей електроприводів, оскільки тепер немає необхідності моделювати кожну функціональну частину системи окремо. Задача проектувальника зводиться до того, щоб вибрати потрібний блок, налаштувати його параметри і підключити до джерела постійної або змінної напруги. У даному розділі розглянемо особливості всіх стандартних моделей електроприводів бібліотеки *Electric Drives Library*, а також їхні загальні властивості і принципи моделювання.

6.3.1 Загальні властивості стандартних моделей електроприводів

Бібліотека *Electric Drives Library* містить три підрозділи: *AC drives*, *DC drives*, *Shafts and speed reducers*, в яких знаходяться відповідно моделі електроприводів змінного струму, постійного струму, а також моделі механічного вала і редуктора. У даному посібнику обмежимося розглядом тільки моделей електроприводів (ЕП) з перших двох підрозділів. Блоки ЕП постійного струму позначені на піктограмах *DC1...DC7*, блоки ЕП змінного струму – *AC1...AC6*. Як приклад на рис. 6.29 показаний зовнішній вигляд блоків *DC1* і *AC1*.

Кожна модель ЕП складається з двигуна постійного чи змінного струму, силового перетворювача і регулятора. Система керування кожного блока містить ПІ-регулятор швидкості (крім блока *AC1*), а також інші типи регуляторів, які докладніше розглянуті в п. 6.3.2 і 6.3.3. Замість регулювання швидкості в системі може бути

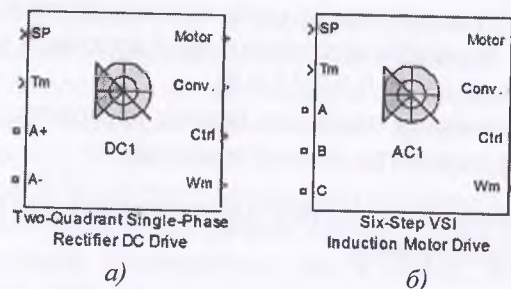


Рисунок 6.29 – Зовнішній вигляд блоків DC1 (а) і AC1 (б)

реалізовано регулювання моменту. Вид керування визначається параметром *Regulation type* на вкладці *Controller* вікна налаштування блока.

6.3.1.1 Два способи моделювання

Для більшості стандартних моделей електроприводів у бібліотеці *Electric Drives Library* можна вибрати один із двох видів моделювання: детальне моделювання (*detailed simulation*) і моделювання в середніх значеннях (*average-value simulation*). У першому випадку для моделі керованого силового перетворювача застосовується блок *Universal Bridge* (див. п. 3.1.1), що дає змогу задати докладний математичний опис випрямляча або інвертора. При цьому для розрахунку моделі програма використовує мале значення кроку, щоб досягти більш точного відображення високочастотних електричних сигналів. При моделюванні в середніх значеннях силовий перетворювач представлений спрощеною математичною моделлю, яка використовує середні значення реальних струмів і напруг. У даному режимі відсутні високочастотні складові сигналів, що дозволяє істотно збільшити значення кроку інтегрування, і, як результат, скоротити час розрахунку всієї моделі. Спрощена модель для кожного стандартного AC- чи DC-блока наведена в довідниковій системі MATLAB до даного блока, що викликається натисканням кнопки *Help* у його вікні налаштування. Там же наводяться рекомендації з вибору значення кроку. Переключення між двома режимами здійснюється установкою параметра *Model detailed level* у значення *Detailed* чи *Average* у вікні налаштування кожної моделі.

6.3.1.2 Вікно налаштування параметрів

Всі стандартні блоки бібліотеки *Electric Drives Library* являють собою досить складні моделі, що складаються з декількох підсистем – двигуна, перетворювача, системи керування. Кожна зі складових частин потребує окремого налаштування. Тому вікно таких моделей містить більшу кількість розділів і полів для введення параметрів у порівнянні з іншими блоками пакета *SPS*. Розглянемо деякі загальні властивості, характерні для всіх моделей електроприводів. Як приклад на рис. 6.30 наведене вікно налаштування блока *DC1 Two-Quadrant Single-Phase Rectifier DC Drive*, який є моделлю електропривода постійного струму з однофазним керованим випрямлячем. Вікно кожного блока містить три основних розділи:

- 1) *Machine* (DC, *Asynchronous* чи *Synchronous*, у залежності від типу двигуна);
- 2) *Converter*;
- 3) *Controller*.

Розділ *Machine* призначений для введення параметрів, які властиві конкретному типу електродвигуна. Ці параметри вже розглядалися в даному посібнику при описі моделей двигунів постійного і змінного струму (п. 6.1.2, 6.1.3). Для зручності вікно розділене на дві частини – електричні (*Electrical*) і механічні (*Mechanical*) параметри. Усі значення задаються в одиницях системи СІ.

Розділ *Converter* описує силовий перетворювач. Сюди вводяться параметри силових ключів (розглянуті в п. 3.1.1) і ланки постійного струму (*DC bus*), якщо вона присутня в даному типі перетворювача. Для ЕП постійного струму задаються також параметри кола збудження двигуна (*Excitation circuit*).

Розділ *Controller* призначений для задання параметрів системи керування електроприводом. Цей розділ зображений у вікні на рис. 6.30. Поле *Regulation type* дає змогу вибрати спосіб керування – за швидкістю (*Speed regulation*) або моментом (*Torque regulation*). У нижній частині вікна розташовані опції для вибору режиму моделювання *Model detail level* (детального чи в середніх значеннях) і механічного входу моделі *Mechanical input* (момент *Tm* або швидкість *w*). При натисканні кнопки *Schematic* відкривається вікно зі схемами регуляторів.

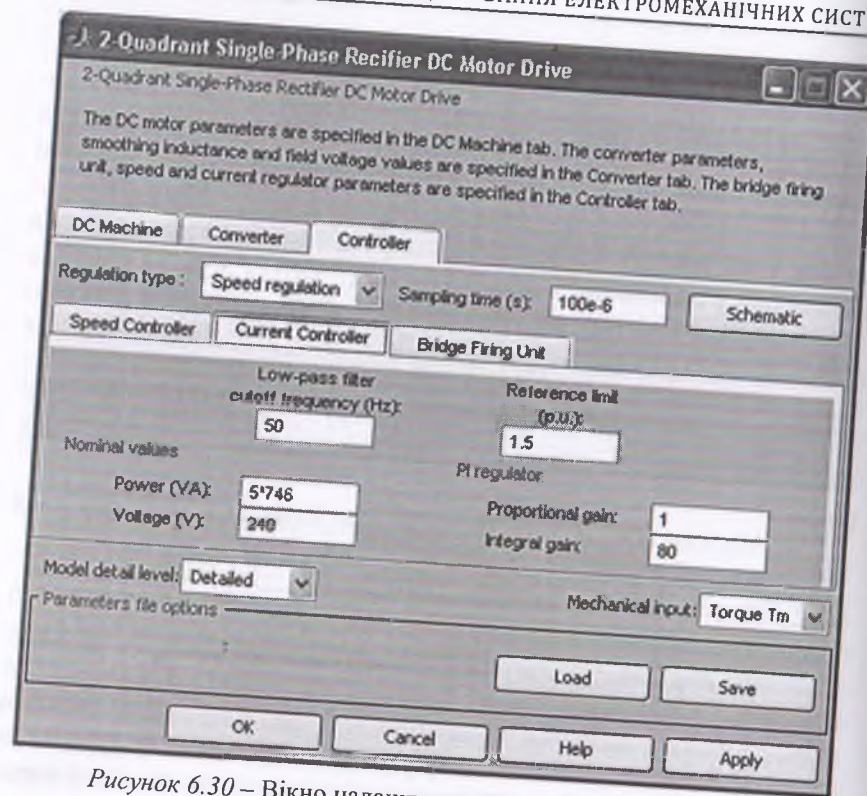


Рисунок 6.30 – Вікно налаштування параметрів блока DC1

Кнопка *Load*, розташована в нижній частині вікна, дає змогу завантажити параметри електропривода з файлу, а кнопка *Save* – зберегти набір використовуваних параметрів у власному файлі з заданим іменем.

Специфічні параметри, що властиві конкретним типам електроприводів постійного і змінного струму, коротко розглянуті у відповідних параграфах даного розділу (див. п. 6.3.2 і 6.3.3).

6.3.1.3 Входи і виходи моделей

Розглянемо призначення входів і виходів блоків бібліотек *AC drives* і *DC drives*. *Simulink*-входи і виходи однакові для всіх моделей (див. рис. 6.29):

SP – робоча точка (*set point*). На цей вхід подається бажане значення кутової швидкості (об/хв) або моменту (Нм), яких двигун

досягне по закінченню перехідного процесу. Характер сигналу (швидкість або момент) залежить від обраного способу керування електроприводом і визначається параметром *Regulation type* на вкладці *Controller* вікна налаштування блока (див. рис. 6.30). У моделях AC1 і AC2 цей параметр відсутній, тому на їхній вхід *SP* подається тільки бажана швидкість;

Tm або *Wm* – вхід для подачі механічного впливу на двигун у вигляді моменту *Tm* або швидкості *Wm*. Визначається параметром *Mechanical input* вікна налаштування (див. рис. 6.30);

Wm або *Te* – вихід для виміру швидкості двигуна *Wm* (якщо на механічний вхід був поданий момент *Tm*), або електромагнітного моменту *Te* (якщо на вхід подавалася швидкість *Wm*);

Motor – векторний сигнал, що містить змінні, які вимірюються на виході двигуна (в одиницях системи СІ);

Conv – векторний сигнал на виході силового перетворювача;

Ctrl – векторний сигнал на виході системи керування;

SPS-порти для включення блока у віртуальне електричне коло розрізняються в залежності від типу перетворювача, використовуваного в даній моделі електропривода:

A+, *A-* – порти для підключення до однофазного джерела синусоїдної напруги (для блоків *DC1*, *DC2*, див. рис. 6.29, а);

A, *B*, *C* – порти для підключення до трифазного джерела (для блоків *DC3*, *DC4*, *AC1–AC6*, див. рис. 6.29, б);

Vcc, *Gnd* – порти для підключення до джерела постійної напруги (для блоків *DC5*, *DC6*, *DC7*).

В усіх випадках напруга джерела має відповідати номінальній напрузі двигуна, що задана у полі *Voltage* в розділі *Controller/Current Controller* для *DC*-моделей (див. рис. 6.30), або в розділі *Asynchronous Machine* – для *AC*-моделей.

6.3.2 Електропривод постійного струму

Підрозділ *DC Drives* бібліотеки *Electric Drives Library* містить 7 базових моделей електроприводів постійного струму (табл. 6.1). За типом перетворювача вони поділяються на дві основні групи: блоки *DC1–DC4* складені на основі мостового керованого випрямляча (КВ), блоки *DC5–DC7* – на основі імпульсного перетворювача постійної напруги. Внутрішню модель кожного блока

і докладний опис англійською мовою можна знайти в довідковій системі програми.

Розглянемо як приклад блок *DC1*, піктограма якого зображена на рис. 6.29, а, вікно налаштування – на рис. 6.30, а функціональна схема його моделі – на рис. 6.31. За допомогою перемикача *SW* (параметр *Regulation type* на вкладці *Controller* вікна налаштування) вибирається один із двох видів керування – за швидкістю або за моментом. У першому випадку (*Speed regulation*) система керується двома ПІ-регуляторами – швидкості (*PI Speed Controller*) і струму (*PI Current Controller*). В другому випадку (*Torque regulation*) регулятор швидкості ігнорується, і в колі керування залишається тільки регулятор струму, при цьому як завдання на вхід системи керування надходить обертаючий момент *Torque reference*, що потім перетворюється в сигнал струму *Current reference*. Для інших блоків бібліотеки *DC Drives* система керування реалізована аналогічно.

Розглянемо коротко основні поля вікна налаштування блоків *DC Drives*. На рис. 6.30 вікно блока *DC1* відкрите на вкладці *Controller*. Цей розділ складається з трьох підрозділів:

- 1) *Speed Controller* – задаються параметри регулятора швидкості (пропорційна й інтегральна складові), номінальна швидкість двигуна *Nominal speed*, початкове задання за швид-

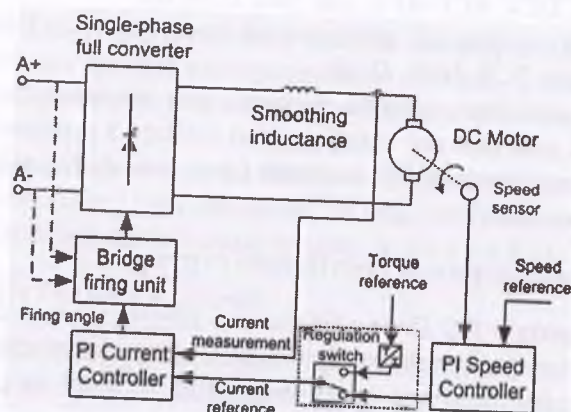


Рисунок 6.31 – Функціональна схема блока *DC1*

кістю *Initial speed reference*, частота зрізу фільтра нижніх частот *Cutoff frequency*, максимально припустимі значення прискорення (*Acceleration*) й уповільнення (*Deceleration*) двигуна, перевищення яких може викликати перевантаження за струмом;

- 2) *Current Controller* (зображений на рис. 6.30) – задаються параметри регулятора струму, номінальні потужність і напруга двигуна *Nominal values*, частота зрізу фільтра, сигнал задання за струмом *Reference limit*; для блоків *DC5–DC7* додатково задається несуча частота для ШІМ-керування силовим IGBT-ключем *PWM Switching Frequency*;
- 3) *Bridge firing unit* – для блоків *DC1–DC4* задаються максимальне (*Alpha max*) й мінімальне (*Alpha min*) значення кута відмикання тиристорів, параметри генератора синхроімпульсів (частота *Frequency* і ширина *Pulse width* імпульсу); для блоків *DC5–DC7* цей розділ відсутній.

Розділ *DC Machine* містить ті ж поля, що і вікно блока *DC Machine* (див. п. 6.1.2, рис. 6.5). Розділ *Converter* для блоків *DC1, DC3* визначає параметри тиристорів мостової схеми (див. п. 3.1.1), для блоків *DC2, DC4* – параметри двох мостових схем (*Converter1* і *Converter2*), оскільки в цих моделях використовується реверсивний випрямляч, а для блоків *DC5, DC6, DC7* – параметри транзистора IGBT, який в імпульсному перетворювачі виступає в ролі керованого ключа. Крім того, у даному розділі для всіх блоків задається значення згладжу вальної індуктивності *Smoothing inductance*, яка включена послідовно з якорем двигуна (рис. 6.31), і напруга джерела живлення для обмотки збудження *Field DC source*.

Сигнали на вимірювальних виходах *DC*-моделей можна одержати за допомогою блока *Bus Selector*. На виході *Motor* вимірюються ті ж змінні, що і на виході *m* блока *DC Machine* (див. п. 6.1.2), а також напруга якоря. На виході *Conv* вимірюється вихідна напруга перетворювача *Vdc*. Вихід *Ctrl* містить чотири елементи: завдання за струмом якоря *Ia*, значення кута комутації на виході регулятора струму *Alpha* (у градусах), похибка регулювання за швидкістю або за моментом *Error* (різниця між заданим і фактичним значеннями сигналу), задання за швидкістю або за моментом *Reference*. Усі сигнали вимірюються у відповідних одиницях системи СІ.

Таблиця 6.1 – Стандартні моделі електроприводів бібліотеки DC Drives

Позначення	Назва блока	Перетворювач
DC1	Two-Quadrant Single-Phase Rectifier DC Drive	Однофазний нереверсивний мостовий КВ
DC2	Four-Quadrant Single-Phase Rectifier DC Drive	Однофазний реверсивний мостовий КВ
DC3	Two-Quadrant Three-Phase Rectifier DC Drive	Трифазний нереверсивний мостовий КВ
DC4	Four-Quadrant Three-Phase Rectifier DC Drive	Трифазний реверсивний мостовий КВ
DC5	One-Quadrant Chopper DC Drive	Імпульсний знижувальний перетворювач (buck converter)
DC6	Two-Quadrant Chopper DC Drive	Імпульсний інвертуючий перетворювач (buck-boost converter)
DC7	Four-Quadrant Chopper DC Drive	Широтно-імпульсний перетворювач (ШІП)

На рис. 6.32 як приклад показана модель електропривода постійного струму на основі блока DC1. У моделі DC1 як силовий перетворювач використовується однофазний тиристорний випрямляч (табл. 6.1), підключений до синусоїдного джерела 220 В через підвищувальний трансформатор. У налаштуваннях блока DC1 встановлено двигун постійного струму потужністю 5 к.с. (3,73 кВА) з номінальною частотою обертання 1750 об/хв і напругою 240 В.

На вхід SP надходить сигнал задання частоти обертання (1000 об/хв), на вхід Tm – момент навантаження, що спочатку дорівнює 10 Нм, а в момент часу $t = 6$ с стрибком змінюється до 25 Нм.

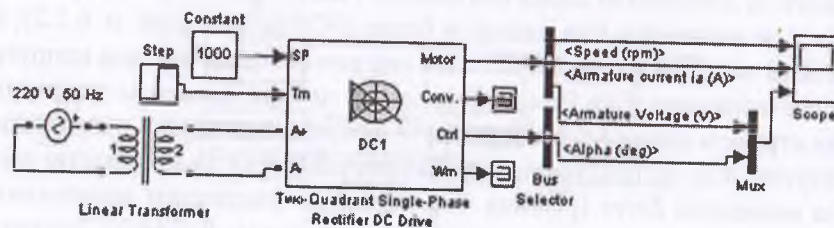


Рисунок 6.32 – Модель ЕП постійного струму на основі блока DC1

Параметр *Regulation type* встановлений у значення *Speed regulation*, отже, система керування складається з двох ПІ-регуляторів – швидкості (РШ) і струму (РС). За заданим значенням швидкості РШ формує сигнал задання по струму, на підставі якого РС розраховує кут керування тиристорами. У результаті напруга на якорі змінюється відповідно до необхідного значення струму.

На рис. 6.33 показані графіки перехідних процесів у моделі: частота обертання двигуна n , струм якоря i_a , напруга u_a , кут керування α . В даному випадку моделювання виконується в середніх значеннях (див. п. 6.3.1.1), тому на графіках струму і напруги відсутні пульсації. Видно, що швидкість досягає заданого значення 1000 об/хв за час $t = 4$ с. Кут керування α упродовж всього процесу залишається меншим за 90° , що відповідає роботі перетворювача у випрямному режимі. При накиданні навантаження змінюється струм якоря відповідно до нового значення навантажувального моменту і трохи підвищується напруга, завдяки чому швидкість двигуна зберігає своє колишнє значення 1000 об/хв.

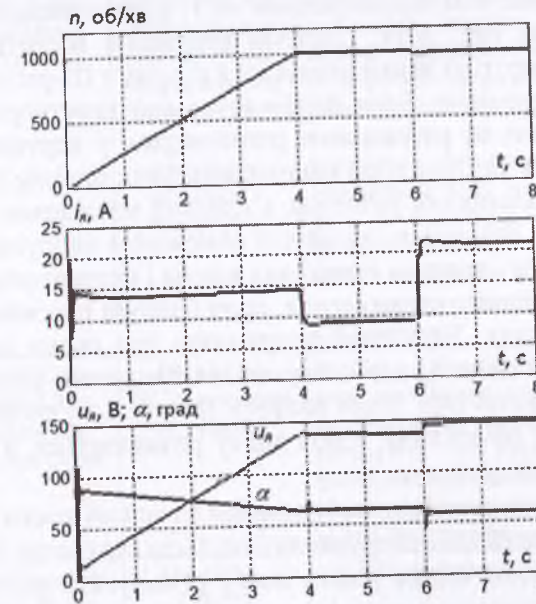


Рисунок 6.33 – Графіки перехідних процесів в ЕП постійного струму (блок DC1)

6.3.3 Електропривод змінного струму

Стандартні моделі електроприводів змінного струму знаходяться в бібліотеці *AC Drives*. Усього в ній 6 моделей, що розрізняються за типом двигуна і способом керування перетворювачем. Усі моделі містять перетворювач частоти (ПЧ) з ланкою постійного струму, причому в деяких моделях до складу ПЧ входить керований випрямляч (КВ), а в деяких – некерований (НВ). Блоки *AC1–AC4* складені на основі асинхронного двигуна (АД) з короткозамкненим або з фазним ротором, блок *AC5* – на основі вентильного синхронного двигуна (ВСД) з фазним ротором, а блок *AC6* використовує синхронний двигун з постійними магнітами (СДПМ). Система керування електроприводом блоків *AC2–AC6* містить ПІ-регулятор швидкості (ПІ-РШ). Особливості систем керування всіх моделей коротко розглянуті в табл. 6.2. Їхній докладний опис англійською мовою, а також функціональні схеми наводяться в довідковій системі програми MATLAB, що відкривається натисканням кнопки *Help* у вікні налаштування кожного блока.

Як приклад розглянемо модель *AC1*, функціональна схема якої зображена на рис. 6.34. Система керування містить зворотний зв'язок за напругою ланки постійного струму з ПІ-регулятором *DC bus voltage regulator*. Блок *Bridge firing unit* перетворює значення кута керування α , розраховане регулятором, у керуючі імпульси, що подаються на тиристори випрямляча. Блок *Braking Chopper* містить у собі гальмівний резистор, з'єднаний послідовно з ключем, і блоки логіки. Він призначений для обмеження напруги в ланці постійного струму шляхом замикання ключа і переведення двигуна в режим динамічного гальмування, коли напруга перевищує верхній поріг обмеження. Тиристори випрямляча при цьому повністю закриваються, і енергія, яка повертається двигуном, розсіюється на гальмівному резисторі. Коли напруга знижується, досягаючи нижнього порогу обмеження, ключ знову розмикається, а випрямляч відновлює свою роботу.

Блок *Six-step generator* побудований на основі шести компараторів і призначений для генерування імпульсів керування інвертором. Додаткові логічні блоки дають змогу реалізувати режим реверса шляхом переключення двох фаз статора. Відкрити внутрішню модель блока *AC1* можна за допомогою параметра *Look under mask* контекстного меню.

Таблиця 6.2 – Стандартні моделі електроприводів бібліотеки *AC*

Позначення	Назва блока	Електропривод	Система керування
AC1	Six-Step VSI Induction Motor Drive	ПЧ (КВ) – АД	ПІ-регулятор напруги ланки постійного струму, що керує випрямлячем. Зворотний зв'язок за швидкістю відсутній (розімкнута система). Реалізується закон $U/f = \text{const}$
AC2	Space Vector PWM VSI Induction Motor Drive	ПЧ (НВ) – АД із ШІМ	ПІ-РШ, що керує ковзанням двигуна в замкнутій системі. Реалізується закон $U/f = \text{const}$
AC3	Field-Oriented Control Induction Motor Drive	ПЧ (НВ) – АД, векторне керування (керування по полю)	Значення моменту і магнітного потоку з виходу ПІ-РШ надходять на вхід векторного регулятора, що змінює струм статора таким чином, щоб виключити взаємний вплив потоків статора і ротора
AC4	DTC Induction Motor Drive	ПЧ (НВ) – АД, DTC-керування (<i>DTC – direct torque and flux control</i> – пряме керування за магнітним потоком і моментом)	ПІ-РШ і DTC-регулятор. Магнітний потік статора й електромагнітний момент розраховуються DTC-регулятором на підставі обмірюваних значень струмів і напруг статора. При цьому потік і момент керуються безпосередньо шляхом порівняння їхніх розрахованих значень з бажаними за допомогою гістерезисних компараторів. Комбінації сигналів на виходах компараторів визначають стан силових ключів інвертора
AC5	Self-Controlled Synchronous Motor Drive	ПЧ (КВ) – ВСД, векторне керування	ПІ-РШ, ПІ-регулятор напруги в колі постійного струму, що керує випрямлячем, а також векторний регулятор, що перетворює вхідні значення струмів, напруг, магнітного потоку і моменту в сигнали керування інвертором
AC6	PM Synchronous Motor Drive	ПЧ (НВ) – СДПМ, векторне керування	ПІ-РШ, векторний регулятор

У моделі AC1 відсутній зворотний зв'язок за швидкістю двигуна, отже, регулювання швидкості не виконується. При цьому бажане значення швидкості, що надходить на вхід SP, використовується тільки для визначення напруги і частоти з метою реалізації закону регулювання $U/f = \text{const}$.

Вікно налаштування параметрів блока AC1 з відкритим розділом *Converters and DC Bus* показано на рис. 6.35. Цей розділ призначений для задання параметрів силового перетворювача.

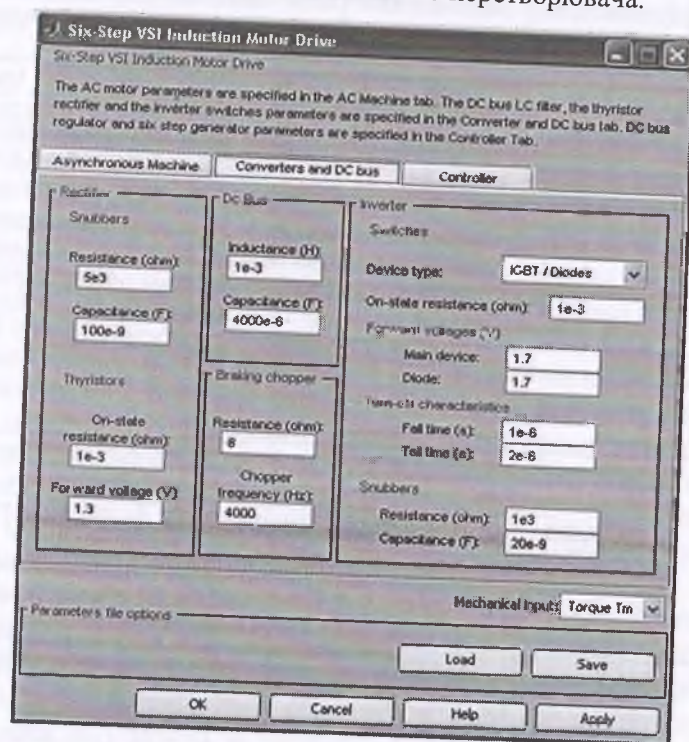


Рисунок 6.35 – Вікно налаштування параметрів блока AC1

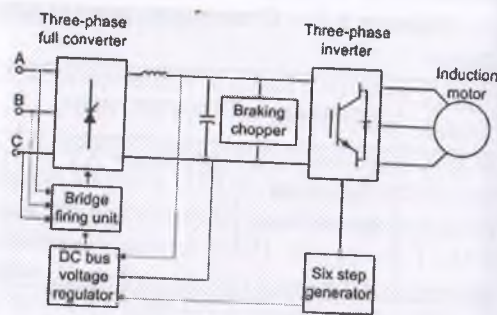


Рисунок 6.34 – Функціональна схема блока AC1

Вікно розділене на 4 поля, які містять параметри основних функціональних частин пристрою: випрямляча (*Rectifier*), інвертора (*Inverter*), ланки постійного струму (*DC Bus*) і гальмівного ключа (*Braking Chopper*). У полях *Rectifier* і *Inverter* задаються ті ж параметри, що й у вікні налаштування блока *Universal Bridge* (п. 3.1.1), у полі *DC Bus* – параметри індуктивно-ємнісного фільтра, а в полі *Braking Chopper* – опір гальмівного резистора і частота замикання ключа.

Вкладки *Asynchronous Machine* і *Controller* призначені для задання параметрів асинхронного двигуна і системи керування відповідно. Їхній докладний опис наведений у довідковій системі.

Сигнали на вимірювальних виходах AC-моделей можна одержати за допомогою блока *Bus Selector*. Вихід *Motor* містить ті ж змінні, що і вихід *m* блока *Asynchronous Machine* (п. 6.1.3). На виході *Conv* вимірюються струми і напруги силових перетворювачів – випрямляча й інвертора. Вихід *Ctrl* містить 3 сигнали: значення кута комутації *Alfa* (у градусах), похибку регулювання за напругою *Voltage Error*, задання за швидкістю *Speed Reference*. Усі сигнали вимірюються у відповідних одиницях системи СІ.

Модель електропривода змінного струму на основі блока AC1 зображена на рис. 6.36. У приводі використовується асинхронний двигун потужністю 3 к.с. (2,24 кВА), що живиться від перетворювача частоти з ланкою постійного струму.

Живлення на вхід системи надходить від трифазного синусоїдного джерела 220 В. Сигнал задання частоти обертання (1800 об/хв) подається на вхід SP блоку AC1, момент навантаження від блока *Timer* – на вхід *Tm*. Графіки перехідних процесів для частоти обертання двигуна n , електромагнітного моменту M_E і напруги U_{dc} у колі постійного струму наведені на рис. 6.37.

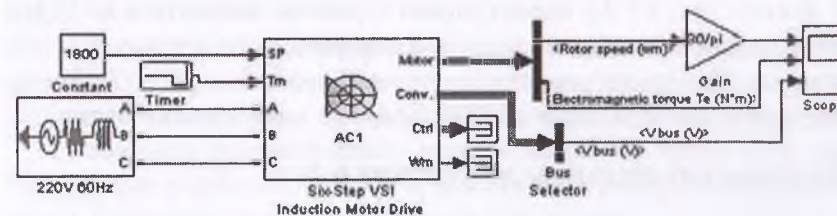


Рисунок 6.36 – Модель ЕП змінного струму на основі блока AC1

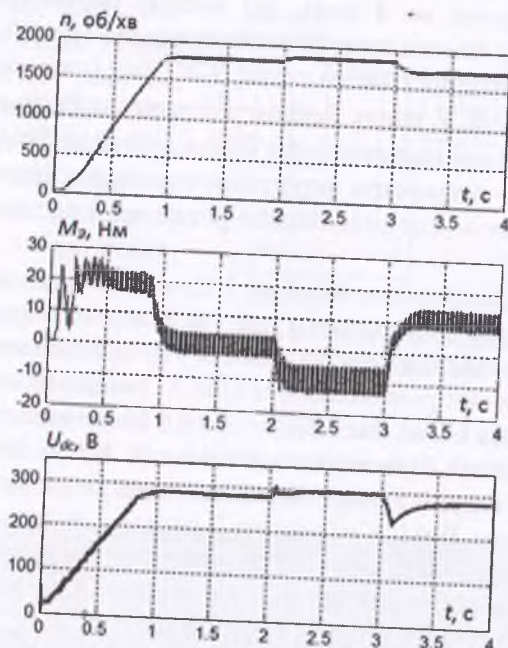


Рисунок 6.37 – Графіки перехідних процесів в ЕП змінного струму (блок AC1)

Пуск двигуна здійснюється без навантаження, і його частота обертання поступово досягає заданого значення 1800 об/хв. У момент часу $t = 2$ с до вала прикладається рушійний момент 10 Нм, і двигун переходить у режим рекуперації, віддаючи енергію в коло постійного струму. Напруга U_{dc} починає зростати, однак регулятор у колі постійного струму (РПС) активізує блок *Braking Chopper*, що зменшує напругу U_{dc} , підтримуючи її практично на постійному рівні. У момент часу $t = 3$ с навантаження стрибком змінюється до 10 Нм. Асинхронна машина переходить з генераторного режиму в режим двигуна. На осцилограмі спостерігається провал напруги U_{dc} , яка потім поступово відновлює своє значення до попереднього рівня.

Контрольні питання до розділу 6

1. Для чого призначений вхід T_m (або TL) у блоків бібліотеки *Machines*?

2. Як реалізувати нелінійну характеристику неробочого ходу для моделей машин змінного струму?
3. Чим відрізняються моделі машин *SI units* і *pu units*?
4. Чим відрізняються моделі синхронних машин: *Simplified Synchr. Mach.*, *Synchr. Mach. Fundamental*, *Synchr. Mach. Standard*?
5. Як реалізувати в моделі роботу синхронної машини в режимі двигуна?
6. З якою метою виконується ініціалізація моделей електричних машин?
7. Для чого призначені стандартні моделі бібліотек *AC drives* і *DC drives* (розділ *Application Libraries /Electric Drives Library*)?
8. Чим відрізняються системи керування при виборі керування за струмом або за моментом в моделях ЕП постійного струму (*DC drives*)?
9. Для чого призначений блок *Braking Chopper* у складі моделей ЕП змінного струму (*AC drives*)?

Завдання для самостійного виконання

1. Скласти модель двигуна постійного струму з незалежним збудженням для дослідження режимів пуску і реверсу. Для зміни полярності постійної напруги на якорі як джерело використовувати блок *Controlled Voltage Source*, керований блоком *Step*.
2. Скласти модель асинхронного двигуна для дослідження режимів прямого пуску і реверсу. Реверс реалізувати за допомогою блоків *Three-Phase Breaker*, що змінюють порядок підключення двох фаз статора.
3. Скласти модель асинхронного двигуна для дослідження режимів прямого пуску і динамічного гальмування (дві фази статора підключаються до джерела постійної напруги через додатковий опір).
4. Скласти модель електроенергетичної системи, зображену на рис. 6.27. Здійснити її ініціалізацію відповідно до опису. Одержати осцилограми перехідних процесів для швидкості і моменту двигуна.
5. Вивчити демонстраційну модель на основі блока *DC6*, що знаходиться у файлі *dc6_example* (ввести ім'я файлу в командному вікні). Пояснити характер отриманих осцилограм.
6. Виконати завдання п. 5 для блока *AC3* (файл *ac3_example*).

Розділ 7

ЗАГАЛЬНІ ПРИНЦИПИ ІДЕНТИФІКАЦІЇ ЕЛЕКТРОТЕХНІЧНИХ ОБ'ЄКТІВ І СИСТЕМ

7.1 Узагальнений алгоритм ідентифікації

На розв'язання задач ідентифікації суттєвий вплив має апріорна (від лат. *apriori* – з попереднього) інформація. Використовується апріорна інформація трьох видів: про фізико-технічні властивості об'єкта; про впливи зовнішнього середовища на об'єкт; про вимірювані змінні.

При вирішенні задач практичної ідентифікації опираються на накопичений досвід, знання та інтуїцію розробників. Узагальнений алгоритм процесу ідентифікації зображено на рис. 7.1.

Ідентифікацію поділяють на ідентифікацію у реальному часі й післяекспериментальну.

У першому випадку на ідентифікацію накладаються жорсткі часові обмеження, у другому – таких обмежень немає, але потрібна більш висока якість моделі. Експериментальна ідентифікація поділяється на два типи: ідентифікація у режимі нормальної експлуатації (пасивний експеримент) і з подачею на об'єкт пробних входних впливів (активний експеримент).

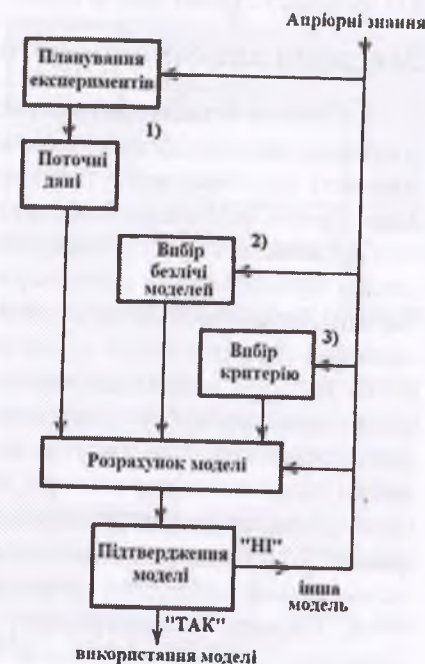


Рисунок 7.1 – Узагальнений алгоритм процесу ідентифікації

Те, що пізнається в результаті обробки вимірів і не залежить від методів і засобів самих вимірів, називається об'єктом ідентифікації. Під об'єктом розуміється все, що пізнається в процесі обробки результатів вимірів або обчислень. Побудова математичної моделі технічного об'єкта або процесу, системи керування технологічним процесом тощо із заданою точністю є завданням ідентифікації. Точність вирішення завдання ідентифікації визначається ступенем адекватності математичної моделі процесам у технологічному об'єкті.

Процес побудови (конструювання) моделей технічних об'єктів за даними спостережень включає в себе три основних компоненти:

- 1) дані;
- 2) моделі-кандидати;
- 3) правила оцінки ступеня відповідності випробуваної моделі, даним спостережень [2].

Прокоментуємо кожний із цих компонентів.

1. *Дані спостережень.* Вхідні-вихідні дані іноді реєструються в процесі проведення цілеспрямованих ідентифікаційних експериментів, коли користувач може визначити перелік і моменти виміру сигналів, причому деякі із вхідних сигналів можуть бути керованими. Завдання планування експериментів, таким чином, полягає у тому, щоб з урахуванням можливих обмежень вибрати максимально інформативні дані про сигнали системи. У деяких випадках користувач не має змоги впливати на хід експерименту й повинен спиратися на дані нормальної експлуатації.

2. *Множина моделей.* Моделі-кандидати устанавлюються за допомогою фіксації тієї групи моделей, у межах якої ми будемо шукати найбільш вдалу модель. Безсумнівно, це найважливіша й у той же час найважча частина процедури ідентифікації. Саме на цьому етапі знання формальних властивостей моделей необхідно з'єднати з апріорним знанням, інженерним мистецтвом, інтуїцією й, безумовно, фізикою процесів, що відбуваються. Безліч моделей іноді стає результатом ретельного моделювання, після чого на основі законів фізики та інших достовірних знань формується модель, що включає в себе фізичні параметри із ще непевними значеннями. Інша можливість полягає в тому, щоб без усякого фізичного обґрунтування використати стандартні, найчастіше – лінійні, моделі. Такі моделі, в яких параметри розглядаються насамперед як варіативні

з підлаштуванням моделей до наявних даних і не відображають фізики процесу, називаються *чорними ящиками*. Безліч моделей з налаштовуваними параметрами, які допускають фізичну інтерпретацію, називають *сірими ящиками*.

3. *Визначення на основі даних спостережень «найкращої» моделі.* Ця частина – *метод ідентифікації*. Оцінка якості моделі пов'язана, як правило, з вивченням поведінки моделей у процесі їхнього використання для відтворення даних вимірів.

Підтвердження моделі. У результаті здійснення всіх трьох етапів процедури ідентифікації одержуємо, хоча б у неявній формі, конкретну модель – одну з безлічі, причому таку, яка відповідно до обраного критерію якості щонайкраще відтворює дані спостережень.

Залишається перевірити, «чи достатньо хороша» модель, тобто чи виконує модель своє призначення. Такі перевірки відомі під назвою *верифікації – процедури підтвердження моделі*. До них належать різні процедури оцінювання відповідності моделей даним спостережень, апіорній інформації і поставленій прикладній меті. Невідповідність моделі кожному із цих компонентів змушує дослідника відмовлятися від обраної моделі, тоді як якісне її функціонування створює певний ступінь довіри до моделі. Модель ніколи не можна вважати остаточною і повним описом системи. Її скоріше можна розглядати як спосіб досить якісного опису тих аспектів поведінки системи, які становлять для дослідника найбільший інтерес.

Контур ідентифікації системи/об'єкта. Процедура ідентифікації системи або об'єкта породжує наступну природну логіку дії (див. рис. 7.1): (1) зібрати дані спостережень (експериментів); (2) вибрати безліч моделей; (3) вибрати найкращу із цієї безлічі моделей-кандидатів. Однак цілком імовірно, що перша з таким чином знайдених моделей не витримає перевірки на етапі її верифікації (підтвердження). Тоді потрібно повернутися й переглянути різні кроки процедури.

Існує кілька причин, з яких модель не відповідає завданню:

- числовий метод не дозволяє знайти найкращу за обраним критерієм модель;
- критерій обраний невдало;
- безліч моделей виявилася неповноцінною у тому розумінні, що в цій безлічі взагалі немає «досить якісного» опису системи (об'єкта);

- безліч даних спостережень не була досить інформативною для того, щоб забезпечити вибір гарних моделей.

Власне кажучи, головним у додатках ідентифікації є ітеративне вирішення всіх цих питань, особливо третього, на основі апіорної інформації і результатів попередніх спроб (див. рис. 7.1). Очевидно, що важливим інструментальним засобом вирішення цього ітеративного завдання є діалогове програмне забезпечення.

Слід зазначити, що незалежно від типу ідентифікації (активна, пасивна) на вхідні сигнали накладається обмеження по інформативності. Так, часові характеристики сигналів інформативні, якщо сам сигнал спільно зі своїми похідними становить лінійно незалежну систему функцій, а число різних частот вхідного сигналу не менше, ніж половина оцінюваних параметрів. При цьому якщо існує хоча б одна частота в частотній характеристиці сигналу помилки виходів об'єкта й моделі і така, що на цій частоті перешкода дорівнює нулю, то завдання ідентифікації може бути вирішене як завгодно точно.

В іншому випадку необхідно визначати частоту, на якій відношення сигнал/перешкода було б максимальним. Точність вирішення завдання ідентифікації в цьому випадку залежить від величини зазначеного відношення. Очевидно, що шукана частота повинна лежати в смузі пропущення частот об'єкта [8].

Методи сучасної технології спрямовані на економічно ефективне виробництво енергії, матеріалів, засобів виробництва і засобів споживання, обробку даних, побудову електромеханічних та інших систем керування. Тому технологія – це сукупність методів обробки, виготовлення, зміни стану, властивостей, форми сировини, матеріалу або напівфабрикату, що реалізуються у процесі сучасного виробництва продукції.

Наука, яка вивчає об'єднання сучасних методів технології з електронікою та механікою називається «механотронікою», основою якої є принцип оптимальності, що забезпечує необхідність одержання найбільшого ефекту при обмежених витратах на технологію, керування нею при наявності різних обмежень, серед яких екологічна чистота, безпека, безаварійність. Таким чином, для цілеспрямованого керування технічним об'єктом дуже важливими є знання про характеристики і властивості об'єкта. Це обов'язкова умова для того, щоб правильно сформований сигнал керування зміг перевести об'єкт із деякого початкового стану в необхідний.

Процес переходу технічного об'єкта (виробничої системи, механізму) у новий стан у більшості випадків здійснюється електромеханічними або механотронними системами.

Отже, ідентифікація систем, будучи цілеспрямованою, повинна складатися з погоджених окремих операцій. Кожна технологічна операція характеризується станом інструмента, блоків, вузлів і механізмів, які далі називаються «об'єктами». Інформація про стан об'єкта необхідна для ухвалення рішення про зміну керуючих впливів на окремі об'єкти або технології в цілому. Звідси – важливість вірогідності й швидкості надходження інформації. Інформацію одержують у вигляді формул, графіків, таблиць шляхом обробки вимірів. Математичні вирази містять інформацію в найбільш концентрованому вигляді. Для стиснення інформації будують математичні моделі, за допомогою яких говорять про стан об'єкта і які застосовують для формування оптимальних, у заданому сенсі, керуючих впливів. Як бачимо вище, істотне значення для побудови математичної моделі має апріорна інформація, яка містить відомості про аналогічні об'єкти, загальні принципи функціонування об'єктів, закони розподілу й числові характеристики перешкод, що діють на об'єкти.

У найзагальнішому вигляді ідентифікований об'єкт (систему керування) можна розглядати як «чорний ящик» (рис. 7.2), що перетворює заданий вплив $g(t)$ у керовану величину $y(t)$.

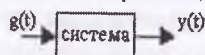


Рисунок 7.2 – Універсальна узагальнена кібернетична модель «black-box» системи (об'єкта)

При такому поданні система описується оператором A , що встановлює однозначний зв'язок між входом і виходом:

$$y(t) = A\{g(t)\},$$

де A – оператор, що підлягає ідентифікації.

7.2 Терміни, визначення, завдання технічної ідентифікації

Більшість термінів, які належать до технічної ідентифікації, також належать до автоматизованого керування технологічними процесами, теорії автоматичного керування й моделювання. Наведемо деякі додаткові.

Об'єкт (у т.ч. керування, регулювання) – технічний пристрій, необхідний режим роботи якого повинен змінюватися (підтримуватися, стабілізуватися) зовнішніми, спеціально організованими впливами.

Елемент – одне з фундаментальних понять загальної теорії систем (системології), математичного моделювання. Походить від латинського «*elementarius*» і означає: початковий, найпростіший, неподільний, той, що лежить в основі чого-небудь. Уперше поняття «елемент» зустрічається в «Метафізиці» Аристотеля.

Система – від грецького «*systema*» – ціле, складене з частин; з'єднання. Безліч елементів, що перебувають у відношеннях і зв'язках одне з одним і утворюють певну цілісність, єдність. Виділяють матеріальні й абстрактні системи. Перші поділяються на системи неорганічної природи і живі. Абстрактні системи – поняття, гіпотези, теорії, наукові знання, формалізовані, логічні та ін.

Стан системи – мінімальний набір змінних величин, здатних однозначно визначити положення системи в цей момент часу.

Стан об'єкта керування – це набір параметрів, що характеризують його в кожен момент часу.

Під *змінними стану* розуміють мінімальну кількість внутрішніх змінних об'єкта, за допомогою яких описуються вихідні змінні. Змінні стану мають важливі властивості, що визначає їхнє широке застосування для опису динамічних процесів і систем. Вхідний сигнал визначає зміни всіх змінних об'єкта у часі, у тому числі й вихідних змінних. Визначення змінної в момент часу t_0 може вимагати зміни цієї змінної на інтервалі $(-\infty, t_0)$. Серед внутрішніх змінних існують такі, що якщо відоме їхнє значення в момент часу t_0 , то цього досить, щоб інші змінні можна було обчислити через відомі і вхідну змінну на інтервалі (t_0, t) . Ці змінні і називаються змінними стану.

Під *структурою* (структурною схемою) системи або об'єкта розуміють сукупність елементів і зв'язків між елементами, обумовлених функціональним і цільовим призначенням системи, об'єкта або їхніх моделей.

Елементом структурної схеми називається найпростіший функціональний перетворювач (оператор), що реалізує різні математичні, логічні та інші операції.

Зв'язком структури називається функціональний перетворювач (оператор) нульового порядку, що має властивості безінерційності,

лінійності, односпрямованості, не має пам'яті й запізнювання. За допомогою зв'язків з'єднуються елементи структури між собою й і з вхідними сигналами. Окремі зв'язки визначають виходи об'єкта або моделі.

Моделлю називається функціональний перетворювач (оператор) певної структури, що відображає властивість (властивості) системи або об'єкта із заданим ступенем адекватності фізичному об'єкту.

Сукупність елементів і зв'язків утворює змінні з коефіцієнтами. Ці коефіцієнти називають параметрами моделі. На відміну від параметрів моделі існують параметри структури.

Об'єкти ідентифікації класифікуються за своїми властивостями або класифікаційними ознаками. Серед таких ознак виділимо наступні: статичність і динамічність у часі, лінійність і нелінійність за вхідним сигналом і параметрами, запізнювання і його відсутність, розподільність і зосередженість, детермінованість і стохастичність структури та параметрів. Окремим класом виділимо об'єкти стійкі й нестійкі, зі швидко і повільно згасаючими коливаннями. Варто також розрізняти об'єкти за кількістю входів і виходів, оскільки існують об'єкти з декількома входами й одним або декількома виходами.

Для побудови алгоритму вирішення завдання ідентифікації важливим є місце внесення перешкоди до об'єкта (до вихідного або вхідного сигналу або до його внутрішньої точки). Безумовно, перешкоди можуть застосовуватися до всіх названих місць одночасно.

Приступаючи до вивчення об'єкта, необхідно знати про існування принципу «чорного ящика», який говорить: якщо на вхід об'єкта подавати як завгодно багато сигналів що відрізняються, і спостерігати як завгодно довго реакцію на ці сигнали, то можна як завгодно точно вивчити поведінку об'єкта. Принцип «чорного ящика» широко застосовується для дослідження об'єктів, особливо на першому етапі пізнання. Однак якщо про об'єкт стає що-небудь відомо, то необхідність у цьому принципі відпадає й варто вибрати більш інформативний метод.

Існує велика кількість об'єктів, для яких характерною є однакова реакція на однакові вхідні впливи. З погляду спостерігача за вхідним і вихідним сигналами такі об'єкти нерозрізнені й називаються *ізоморфними*.

Як приклад ізоморфних об'єктів можуть служити термометри різних конструкцій і принципів дії для виміру температури в тому самому діапазоні її зміни.

Другим прикладом ізоморфізму служать механічна й електрична системи – маятник та електричний *LC-контур*, коли перший, будучи виведеним зі стану рівноваги, повертається в нього, а другий заряджає конденсатор після замикання контуру. При цьому вихідними величинами служать кут відхилення маятника φ і напруга на конденсаторі U_c . Такі об'єкти називають *об'єктами-аналогами*.

Третім прикладом може служити будь-який об'єкт і його голограма – зображення об'єкта, засноване на інтерференції хвиль. Голограма відображає тільки одну властивість об'єкта – його зовнішній вигляд.

З наведених прикладів видно подібність об'єктів. Так, у першому випадку об'єкти нерозрізнені з погляду виконуваних ними функцій, у другому – характеризуються аналогічною реакцією на аналогічні впливи, у третьому – мають зовнішню подібність. Важливим тут є те, що за властивостями одного об'єкта можна вивчати деякі властивості іншого, тобто один об'єкт є моделлю-заступником іншого.

Спрощеною, або *гомоморфною*, моделлю називають таку модель, яка однозначно характеризує тільки деякі риси об'єкта. Клас гомоморфних моделей дуже широкий. Це пояснюється тим, що спрощені моделі застосовують у вигляді деяких математичних виразів, які описують характер поведінки об'єкта. Такі моделі називають математичними.

Як визначено, об'єктом ідентифікації є все те, що пізнається за інформацією, одержуваною в процесі обробки даних. Дані одержують шляхом вимірів змінних на деякому кінцевому інтервалі спостереження, а також шляхом обчислень відомих функцій, що задають аналітично або алгоритмічно. Для цілей ідентифікації прагнуть одержати інформацію в найбільш стислому вигляді, що досягається шляхом побудови математичних моделей. При цьому будують математичні моделі двох типів.

Перший тип – математичні моделі, які максимально докладно описують внутрішні і зовнішні процеси, що проходять в об'єкті, за допомогою яких ніби імітують об'єкт. Такі математичні моделі називають імітаційними.

Другий тип – математичні моделі, які відповідають обмірюваним або обчисленим змінним, що дозволяє одержати прогнозне значення об'єкта із заданою точністю. Такі моделі відрізняє, як правило,

простота опису в порівнянні з імітаційними моделями. Надалі будемо розглядати тільки математичні моделі другого типу. Цей тип математичних моделей будують виходячи з наступних міркувань.

Вихідний сигнал досліджуваного лінійного об'єкта можна представити у вигляді суми двох складових – вільної і вимушеної:

$$Y_o(t) = Y_v(t) + Y_z(t).$$

Вільна складова характеризує перехідний процес, а отже, вагову функцію $K(t)$, яку можна записати як реакцію досліджуваного об'єкта на вхідний сигнал типу дельта-функції $\delta(\tau)$, тобто $K(t) = Y_v(\delta(\tau), t)$, де τ визначає момент доповнення $\delta(\tau)$ до входу досліджуваного об'єкта.

Вагова функція $K(t)$ є розв'язком відповідного диференціального або інтегрального рівняння, може задаватися в безперервній і дискретній формі. Тому математична модель вільної складової може мати різні форми запису, що відрізняються як структурою, так і параметрами. Вибір тієї або іншої математичної моделі залежить від розв'язуваного дослідником завдання та його індивідуальних особливостей. Вимушена складова $Y_v(t)$ являє собою реакцію досліджуваного об'єкта в сталому режимі, тобто в такому режимі, коли перехідний процес скінчився, а зміна вихідного сигналу відбувається тільки під дією вхідного сигналу. При дослідженні сталого режиму роботи для об'єкта визначається функція часу $Y_{ст}(t) = P \cdot U_{вх}(t)$, яка характеризує процес відстеження об'єктом вхідного сигналу $U_{вх}(t)$ конкретної форми при параметрі об'єкта P . Для нелінійних об'єктів дослідження значно складніше.

Так, розділити вихідний сигнал на вільну й вимушену складові не завжди можливо, тому доводиться розглядати ці системи спільно. Однак часто зустрічаються об'єкти, коли вільною складовою можна знехтувати, і тоді розглядається тільки вимушена складова. Розгляд характеристик «вхід-вихід» у параметричному вигляді, тобто залежностей $Y_v = F(U_{вх})$, також не вимагає поділу вихідного сигналу на складові. У цьому випадку будується математична модель не для вхідного сигналу конкретної форми, а залежно від амплітуди вихідного сигналу.

Очевидно, що крім амплітуди певну роль відіграє швидкість, прискорення й інші характеристики вхідного сигналу. Практичні потреби в описі об'єктів породжують розмаїтість класів моделей.

З урахуванням викладеного уточнимо основні завдання ідентифікації (рис. 7.3). Під ідентифікацією динамічних об'єктів будемо розуміти процедуру визначення структури і параметрів їхніх математичних моделей, які при однакових вхідних сигналах на об'єкт і на модель забезпечують близькість вихідних сигналів моделі до вихідних сигналів об'єкта при наявності певного критерію оцінки близькості – критерію якості ідентифікації.

Основні етапи ідентифікації наступні.

1. Етап структурної ідентифікації полягає у визначенні структури математичної моделі на підставі теоретичних міркувань й апріорної інформації.

2. Етап параметричної ідентифікації містить у собі проведення ідентифікаційного експерименту з наступним визначенням параметрів моделі.

3. Етап перевірки адекватності – перевірка якості моделі у сенсі обраного критерію близькості виходів моделі й об'єкта (верифікація моделі).

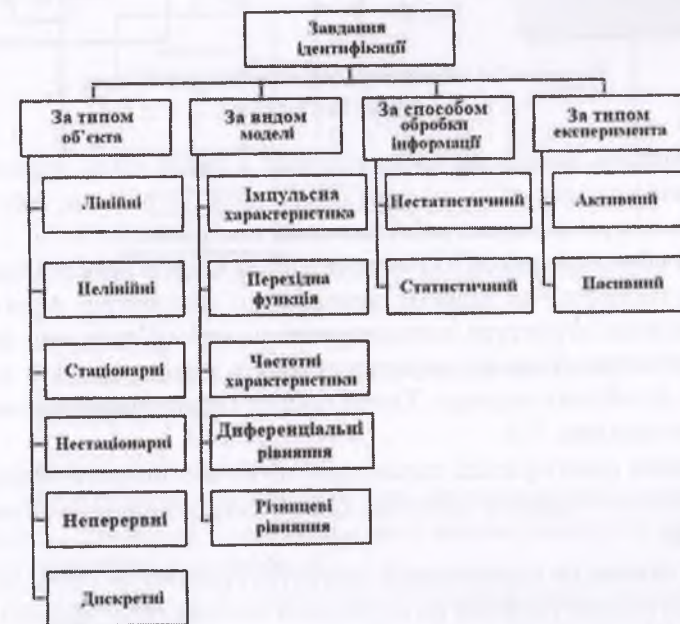


Рисунок 7.3 – Класифікація завдань ідентифікації

7.3 Методи вирішення завдань ідентифікації параметрів математичних моделей

Як зазначалося раніше, ідентифікація динамічних об'єктів у загальному випадку полягає у визначенні їхньої структури і параметрів за спостережуваними даними – вхідним впливом і вихідними величинами. У цьому випадку об'єкт (елемент системи, ланка, об'єкт керування, елемент технологічного процесу тощо) являє собою «чорний ящик». Необхідно, піддаючи об'єкт зовнішнім впливам й аналізуючи його реакції, одержати математичну модель (опис його структури і параметрів), тобто перетворити «чорний ящик» у «білий ящик», домогтися його «інформаційної прозорості». Графічно процес ідентифікації ілюструє рис. 7.4.

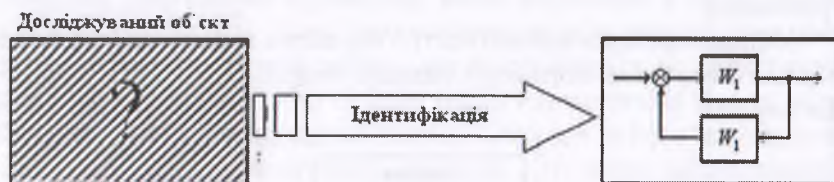


Рисунок 7.4 – Ідентифікація структури об'єкта (у «широкому» сенсі)

Важливим моментом цього процесу є вибір місць доповнення зовнішніх впливів і збір інформації про реакцію об'єкта, тобто розміщення керуючих пристроїв і датчиків системи.

При ідентифікації об'єктів вирішується і більш просте (відносно просте) завдання – це завдання ідентифікації параметрів, коли заздалегідь відома структура математичної моделі об'єкта, але невідомі її параметри. У цьому випадку говорять про перехід від «сірого ящика» до «білого ящика». Такий процес ідентифікації параметрів ілюструється рис. 7.5.

Завдання ідентифікації параметрів може або входити компонентом у загальне завдання ідентифікації об'єкта, або вирішуватися самостійно.

Розглянемо на узагальненій структурі і процедурі процесу ідентифікації основи підходів до вирішення завдань ідентифікації.

Узагальнена структура процесу ідентифікації показана на рис. 7.6.

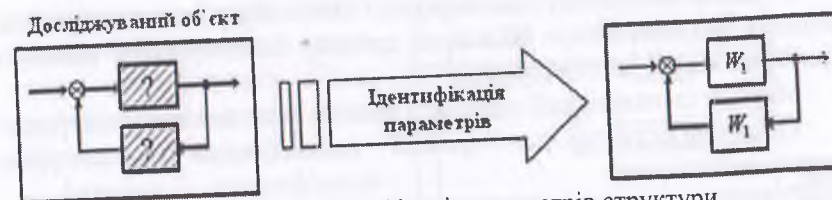


Рисунок 7.5 – Ідентифікація параметрів структури (у «вузькому» сенсі)

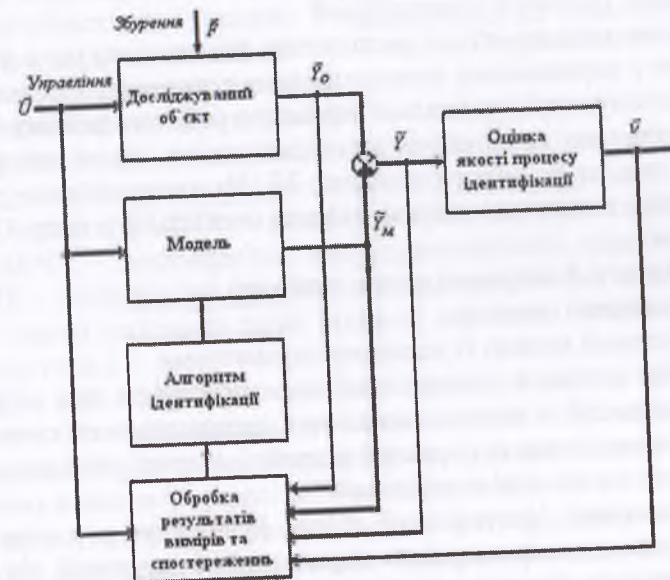


Рисунок 7.6 – Узагальнена структурна схема процедур ідентифікації об'єкта

7.4 Узагальнена структурна схема процесу ідентифікації

1. Класифікація об'єкта.
2. Вибір для певного класу об'єкта налаштовуваної моделі, тобто моделі, структуру й параметри якої можна змінювати (налаштовувати) в процесі ідентифікації.
3. Вибір критерію (оцінки) якості ідентифікації, що характеризує у вигляді функціонала доступних для спостереження змінних відмінність моделі й об'єкта.

4. Вибір алгоритму ідентифікації (механізму налаштування моделі), що забезпечує збіжність процесу ідентифікації, мінімум критерію якості ідентифікації.

Методи ідентифікації прийнято розділяти на дві наступні групи:

1) методи активної ідентифікації – ідентифікація поза контуром керування;

2) методи пасивної ідентифікації – ідентифікація в контурі керування.

Методи активної ідентифікації

У цьому випадку об'єкт дослідження виводиться з умов функціонування у нормальному навколишньому середовищі (нормальний режим експлуатації, номінальні параметри робочого режиму і т. п.).

Дослідження проводяться в спеціалізованих, часто лабораторних, умовах, як це показано на рис. 7.7. На входи об'єкта (робочі і додаткові) подаються тестові сигнали спеціального виду. Це можуть бути:

- а) східчасті й імпульсні часові сигнали;
- б) гармонічні сигнали;
- в) випадкові впливи із заданими параметрами.

Методи активної ідентифікації використовують при розробці нових технологій відповідно до діючих промислових об'єктів, при вивченні нових явищ та первісній розробці математичної моделі.

Методи пасивної ідентифікації

При пасивній ідентифікації об'єкт функціонує в контурі керування, знаходиться в процесі нормальної експлуатації. На його входи надходять тільки природні сигнали керування. Пасивну ідентифікацію використовують для уточнення математичної моделі та спостереження за змінами в об'єкті. Інформація оперативно використовується у системі керування об'єктом, процес такої ідентифікації ілюструється рис. 7.8.

Крім перерахованих груп методів, реалізуються і системи ідентифікації змішаного типу, коли об'єкт не виводиться з нормального режиму експлуатації, але

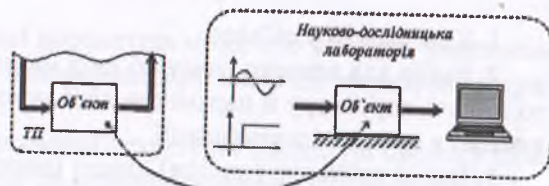


Рисунок 7.7 – Активна ідентифікація у лабораторних умовах

до керуючих сигналів додаються тестові впливи, що дозволяють ідентифікувати об'єкт, не погіршуючи якості основного процесу керування.

Активна ідентифікація об'єктів керування й автоматизації може вироблятися як у часовій області, так і у частотній. При цьому в кожній області використовують власні алгоритми і методи ідентифікації.

При активній ідентифікації в більшості випадків використовують отримані в результаті експериментів характеристики:

- частотні (АФЧХ – амплітудно-фазо-частотні характеристики, ЛАЧХ – логарифмічні амплітудно-частотні характеристики, ЛФЧХ – логарифмічні фазо-частотні характеристики та ін.);
- часові (східчаста зміна задання, «вузький» імпульс задання, меандр та ін.).

При ідентифікації, наступному аналізі та синтезі об'єкта або системи автоматизованого керування потрібен такий математичний опис, який би зв'язував поведінку координат об'єкта/системи – змінних величин у процесі роботи, тобто в часі.

Звісно, що при обробці результатів ідентифікації можливі спрощення, наприклад – лінеаризація. Зміна координат об'єкта/системи у часі називається динамікою. Оскільки система складається зі взаємозалежних функціональних елементів (наприклад, таких, що описуються елементарними ланками ТАК), то для одержання її математичного опису необхідно мати математичні описи окремих елементів. Математичний опис елемента встановлює зв'язок у часі між його поточними значеннями вихідних $y(t)$ і вхідних $x(t)$ величин. Динаміка елемента, тобто поведінку його координат у часі, описується диференціальними рівняннями. У динаміці зв'язок між координатами, тобто між вхідними і вихідними величинами, змінюється у часі. Динаміка характеризується перехідним процесом. При часі, що прагне до нескінченності $t \rightarrow \infty$, поточні координати $y(t)$ і $x(t)$ приймають сталі значення і настає статичний режим. Цей режим описується алгебраїчними рівняннями. У статиці вхідні

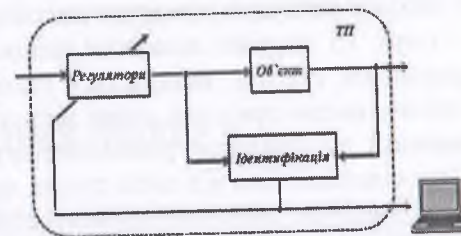


Рисунок 7.8 – Застосування пасивної ідентифікації для налаштування системи керування об'єктом

й вихідні величини елемента постійні: $x(\infty) \equiv x_0 = \text{const}$; $y(\infty) = y_0 = \text{const}$. Ці постійні величини називаються сталими, а процес, що відповідає статиці, називається статичним процесом. Теоретично, статика настає при $t \rightarrow \infty$, однак у практичній ідентифікації прийнято вважати, що статичний режим настає вже тоді, коли поточні координати відрізняються від своїх сталих значень не більше ніж на $\pm 5\%$.

Динамічне рівняння окремого елемента складається за правилами відповідної технічної науки (елемент може являти собою електричну машину, механічну передачу, нагрівальний прилад, електричне коло, електронну схему тощо).

Контрольні питання до розділу 7

1. Поясніть узагальнений алгоритм ідентифікації.
2. Назвіть відомі Вам методи наукового пізнання технічних об'єктів.
3. Що таке ідентифікація у «вузькому» і «широкому» розумінні?
4. Класифікуйте завдання ідентифікації.
5. Які існують математичні моделі об'єктів автоматизації?
6. Як вирішується проблема верифікації моделі?
7. Які існують етапи при моделюванні технічних систем?
8. Що таке імітаційне моделювання у техніці?
9. Що таке «змінні стану системи»?
10. Які існують особливості спостережень і вимірів при моделюванні?
11. Поясніть, що таке емерджентність.
12. Поясніть, що розуміється під стохастикою технологічного процесу.
13. Що позначають терміни «ізоморфізм» і «гомоморфізм» об'єктів?
14. Дайте визначення процесу ідентифікації об'єкта, ідентифікації параметрів.
15. Які методи належать до методів пасивної ідентифікації?
16. Які методи належать до методів активної ідентифікації?
17. Яку інформацію несе фазова частотна характеристика ланки?
18. Перерахуйте основні елементи структурних схем систем керування.
19. Поясніть «принцип чорного ящика»
20. Як проводиться активна ідентифікація у лабораторних умовах?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Герман-Галкин, С. Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0 [Текст]: Учебное пособие / С. Г. Герман-Галкин – СПб.: КОРОНА принт, 2001. – 320 с.
2. Герман-Галкин, С. Г. Линейные электрические цепи: лабораторные работы [Текст] / С. Г. Герман-Галкин – СПб.: КОРОНА принт, 2002. – 192 с.
3. Герман-Галкин, С. Г. Силовая электроника: лабораторные работы на ПК [Текст] / С. Г. Герман-Галкин – СПб.: КОРОНА принт, 2002. – 304 с.
4. Герман-Галкин С. Г., Электрические машины: лабораторные работы на ПК [Текст] / С. Г. Герман-Галкин, Г. А. Кардонов – СПб.: КОРОНА принт, 2003. – 256 с.
5. Герман-Галкин, С. Г. Компьютерное моделирование полупроводниковых систем. Matlab 6 [Текст] / С. Г. Герман-Галкин – СПб.: «Корона принт», 2001. – 298 с.
6. Гульяев, А. Визуальное моделирование в среде MATLAB [Текст] / А. Гульяев – СПб.: Питер, 2000. – 429 с.
7. Дорф, Р. Современные системы управления [Текст] / Р. Дорф, Р. Бишоп – М.: Лаборатория Базовых Знаний, 2004. – 467 с.
8. Дьяконов, В. MATLAB 6: учебный курс [Текст] / В. Дьяконов – СПб.: Питер, 2001. – 592 с.
9. Дьяконов, В. Simulink 4. Специальный справочник [Текст] / В. Дьяконов – СПб.: Питер, 2002. – 528 с.
10. Дьяконов, В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник [Текст] / В. Дьяконов, В. Круглов – СПб.: Питер, 2002. – 448 с.
11. Дьяконов, В. Математические пакеты расширения MATLAB. Специальный справочник [Текст] / В. Дьяконов, В. Круглов – СПб.: Питер, 2001 – 525 с.
12. Лозинський, А. О. Розв'язання задач електромеханіки в середовищах MathCAD і MATLAB: навчальний посібник [Текст] / А. О. Лозинський, В. І. Мороз, Я. С. Паранчук – Львів: Вид-во ДУ «Львівська політехніка», 2000. – 166 с.
13. Медведев, В. С. Control System Toolbox. MATLAB 5 для студентов / Под общ. ред. В. Г. Потемкина [Текст] / В. С. Медведев, В. Г. Потемкин – М.: ДИАЛОГ-МИФИ, 1999. – 287 с.
14. Кортунув, В. И. Моделирование динамических систем в среде Simulink: учеб. пособие по курсовому и дипломному проектированию [Текст] / В. И. Кортунув, И. Ю. Дыбская – Харьков: ХАИ, 1999. – 131 с.
15. Онищенко, О. А. Моделирование электромеханических систем: конспект лекций [Текст] / О. А. Онищенко, О. Я. Карпович – Одесса: Військова академія, 2012. – 136 с. (електронна версія).
16. Современный анализ систем управления с применением MATLAB, Simulink, Control System: учебное пособие [Текст] – К.: «Корнійчук», 1999. – 144 с.

Додаток А

Перелік основних бібліотечних блоків, що входять до складу пакетів *Simulink* та *Sim Power Systems* середовища *MATLAB 7.3.0*.

Назва блока	Призначення	Посилання на підрозділи у тексті посібника
1	2	3
Пакет Simulink		
Бібліотека Sources (Джерела сигналів)		
Constant	Генератор постійного сигналу	п. 1.2.1
Pulse Generator	Генератор прямокутного сигналу	п. 1.2.1, п. 3.2.1.1
Ramp	Генератор лінійно-наростаючого сигналу	п. 1.2.1, п. 1.2.3.1, п. 2.3.1.3, п. 3.2.3.2
Repeating Sequence	Генератор пилкоподібного сигналу	п. 1.2.1, п. 3.2.1.2
Signal Generator	Універсальний генератор сигналу	п. 1.2.1, п. 1.4.2.2
Sine Wave	Генератор синусоїдного сигналу	п. 1.2.1, п. 1.2.3.3
Step	Генератор одиничного перепаду	п. 1.2.1, п. 1.3.2, п. 2.4
Бібліотека Sinks (Приймачі сигналів)		
Display	Цифровий індикатор	п. 1.1.4.2, п. 1.2.2.3
Scope	Осцилограф	п. 1.1.4.1
To Workspace	Блок запису сигналу в робочу область MATLAB	п. 1.1.4.3, п. 2.5.2
XY Graph	Графопобудовник	п. 1.1.4.1, п. 4.1.2
Бібліотека Math Operations (Математичні операції)		
Divide	Ділення	п. 2.7, п. 3.2.1.2
Gain	Підсилювач, пропорційна ланка	п. 1.3.1.4
Product	Добуток	п. 2.3.1.3, п. 3.2.3.2, п. 2.7
Sum	Додавання/віднімання	п. 2.4.2
Trigonometric Function	Тригонометрична функція	п. 2.3.1.3
Бібліотека Continuous (Безперервні елементи)		
Derivative	Диференціююча ланка	п. 1.3.1.5, п. 2.7
Integrator	Інтегруюча ланка	п. 1.3.1.6
State-Space	Модель об'єкта у формі простору станів	п. 1.3.1.3
Transfer Fcn	Передатна функція	п. 1.3.1.1

Продовження дод. А

Продовження дод. А

1	2	3
Transport Delay	Затримка сигналу за часом, ланка із запізнюванням	п. 1.3.1.8
Zero-Pole	Модель об'єкта у формі нулів і полюсів	п. 1.3.1.2
Бібліотека Discontinuities (Нелінійні елементи)		
Hit Crossing	Індикатор перетинання сигналом граничного значення	п. 1.2.3.3
Relay	Реле	п. 1.2.3.2, п. 3.2.1.2
Saturation/Saturation Dynamic	Обмежувач сигналу	п. 1.2.3.1, п. 3.2.3.2
Бібліотека Signal Routing (Маршрутизатори сигналу)		
Bus Selector	Шина	п. 4.1.1, п. 4.1.3, п. 4.3.2
Demux	Демультіплексор	п. 1.2.2, п. 2.2.5.1, п. 2.2.5.3
Multiport Switch	Багатовходовий перемикач	п. 1.2.2.2
Mux	Мультіплексор	п. 1.2.2, п. 1.2.3.3
Selector	Селектор	п. 1.2.2.3, п. 3.2.2.2
Switch	Керований перемикач	п. 1.2.2.1, п. 2.6.2
Бібліотека Ports & Subsystems (Підсистеми)		
Enabled/Triggered Subsystem	Підсистеми, керовані сигналом	п. 1.4.2.1, п. 1.4.2.2
For/While Iterator Subsystem	Підсистеми, керовані в циклі	п. 1.4.2.5, п. 1.4.2.6
If/Switch Case Action Subsystem	Підсистеми, керовані за умовою	п. 1.4.2.4
Subsystem	Некерована підсистема	п. 1.4.1
Бібліотека Model Verification (Блоки верифікації сигналу)		
Check Static/Dynamic Gap	Підтвердження сигналу поза межами статичного/динамічного діапазону	п. 1.2.4
Check Static/Dynamic Range	Підтвердження сигналу у межах статичного/динамічного діапазону	
Check Static/Dynamic Upper Bound	Підтвердження, що сигнал знаходиться нижче верхнього рівня	
Check Static/Dynamic Lower Bound	Підтвердження, що сигнал знаходиться вище нижнього рівня	
Блоки з інших бібліотек пакета Simulink		
Data Type Conversion	Перетворювач типу сигналу (бібліотека Signal Attributes)	п. 1.2.3.3, п. 1.2.4

Продовження дод. А

1	2	3
Fcn	Функціональна залежність (бібліотека User-Defined Functions)	п. 2.6.2
PID Controller	ПІД-регулятор (бібліотека Simulink Extras/Additional Linear)	п. 1.3.1.7, п. 3.2.2.1
Memory	Затримка сигналу на 1 крок моделювання (бібліотека Discrete)	п. 1.4.2.1, п. 1.4.2.2
Пакет SimPower Systems (powerlib)		
Бібліотека Electrical Sources (Джерела електричних сигналів)		
AC Current/Voltage Source	Джерела синусоїдного струму і напруги	п. 2.2.1, п. 3.2.2.2, п. 2.4.
Controlled Current/Voltage Source	Керовані джерела струму і напруги	п. 2.3.1.3, п. 2.5.1, п. 2.6.1, п. 3.1.2
DC Voltage Source	Джерело постійної напруги	п. 2.4
Three-Phase Source	Трифазне джерело напруги	п. 2.3.1.1
Three-Phase Programmable Voltage Source	Трифазне програмоване джерело напруги	п. 2.3.1.2
Бібліотека Elements (Елементи електричних кіл)		
Breaker	Рубильник/автоматичний вимикач	п. 2.2.3, п. 2.4
Linear Transformer	Лінійний однофазний трансформатор	п. 2.2.4
Series/Parallel RLC Branch	Послідовна і паралельна RLC-ланка	п. 2.2.2, п. 2.4
Series/Parallel RLC Load	Послідовне і паралельне RLC-навантаження	п. 2.2.2
Three-Phase Breaker	Трифазний вимикач	п. 2.3.2
Three-Phase Fault	Модель трифазного короткого замикання	п. 2.3.3
Three-Phase Series/Parallel RLC Branch/Load	Трифазна послідовна/паралельна RLC-ланка/навантаження	п. 2.3.2
Three-Phase Transformer	Трифазний трансформатор	п. 2.3.2
Бібліотека Power Electronics (Елементи силових електроніки)		
Diode	Діод	п. 3.1.1
Ideal Switch	Ідеальний ключ	п. 2.2.3, п. 2.4, п. 3.1.1
IGBT	Транзистор IGBT	п. 3.1.1
GTO	Керований тиристор	п. 3.1.1
MOSFET	Транзистор MOSFET	п. 3.1.1

Закінчення дод. А

1	2	3
Thyristor/ Detailed Thyristor	Тиристор (спрощена і повна моделі)	п. 3.1.1
Universal Bridge	Універсальна мостова схема	п. 3.1.1, п. 3.2.1.3
Бібліотека Machines (Електричні машини)		
Asynchronous Machine	Асинхронна машина	п. 4.1.3
DC Machine	Машина постійного струму	п. 4.1.2
Excitation System	Система збудження генератора з регулятором напруги	п. 4.1.4.3
Machines Measurement Demux	Вимірник вихідних сигналів електричних машин	п. 4.1.1
Simplified Synchronous Machine	Спрощена синхронна машина	п. 4.1.4.1
Synchronous Machine Fundamental/ Standard	Синхронна машина (базова і стандартна моделі)	п. 4.1.4.2
Steam Turbine and Governor	Парова турбіна з регулятором частоти обертання	п. 4.2.1
Бібліотека Measurements (Вимірювальні прилади)		
Current/Voltage Measurement	Вимірники струму і напруги	п. 2.2.5.1
Impedance Measurement	Вимірник повного опору (імпедансу)	п. 2.2.6.1
Multimeter	Універсальний вимірник (мультиметр)	п. 2.2.5.1
Three-Phase V-I Measurement	Трифазний вимірник струму і напруги	п. 2.3.4
Бібліотека Extra Library/ Measurements (Додаткові вимірювальні елементи)		
Active & Reactive Power	Вимірник активної і реактивної потужності	п. 2.2.5.3
Fourier	Вимірник амплітуди і початкової фази заданої гармоніки сигналу	п. 2.2.5.4
RMS	Вимірник діючого значення сигналу	п. 2.2.5.2
Бібліотека Extra Library /Control Blocks (Додаткові елементи систем керування)		
PWM Generator	Генератор ШІМ-імпульсів	п. 3.2.3
Synchronized 6-Pulse Generator	Генератор імпульсів із синхронізацією	п. 3.2.2
Timer	Таймер	п. 3.2.1.2
3-phase Programmable Source	Трифазне програмоване джерело сигналу	п. 2.3.1.2, п. 3.2.3.2

Додаток Б

Б1 – Технічні параметри асинхронних двигунів серії 4А

№ з/п	Тип двигуна	Р _{2н'} кВт	S _{н'} %	η _{н'} %	cosφ	Параметри схеми заміщення у номінальному режимі, в.о.				
						X _н	R ₁	X ₂	R ₂ '	X ₂ '
Синхронна частота обертання 3000 об/хв										
1	4AA63B2Y3	0,55	8,5	73,0	0,86	2,5	0,13	0,049	0,049	0,097
2	4A71B2Y3	1,1	6,3	77,5	0,87	2,8	0,13	0,054	0,069	0,084
3	4A80B2Y3	2,2	4,3	83,0	0,87	2,7	0,076	0,050	0,049	0,087
4	4A100S2Y3	4	3,3	86,5	0,89	3,4	0,054	0,055	0,036	0,099
5	4A100L2Y3	5,5	3,4	87,5	0,91	3,8	0,050	0,054	0,036	0,11
6	4A112M2Y3	7,5	2,5	87,5	0,88	3,7	0,046	0,058	0,028	0,14
7	4A160S2Y3	15	2,1	88,0	0,91	4,0	0,052	0,092	0,022	0,12
8	4A180S2Y3	22	1,9	88,5	0,1	3,6	0,039	0,091	0,020	0,11
9	4A180M2Y3	30	1,8	90,5	0,90	3,8	0,030	0,073	0,018	0,11
10	4A200S2Y3	45	1,9	91,0	0,90	4,9	0,027	0,088	0,020	0,13
11	4225M2Y3	55	1,8	91,0	0,92	5,6	0,026	0,092	0,019	0,12
12	4A250S2Y3	75	1,4	91,0	0,89	4,8	0,021	0,080	0,015	0,13
13	4A250M2Y3	90	1,4	92,0	0,90	5,2	0,020	0,078	0,016	0,13
14	4A280S2Y3	110	2,0	91,0	0,89	3,8	0,017	0,097	0,013	0,10
15	4A315S2Y3	160	1,9	90,0	0,90	5,0	0,015	0,093	0,012	0,12
16	4A355S2Y3	250	1,9	92,5	0,90	5,7	0,013	0,091	0,011	0,10
Синхронна частота обертання 1000 об/хв										
17	4A160S6Y3	11	2,7	86,0	0,86	3,0	0,073	0,11	0,030	0,15
18	4A160M6Y3	15	2,6	87,5	0,87	3,0	0,062	0,10	0,028	0,16
19	4A200M6Y3	22	2,3	90,0	0,90	4,1	0,050	0,11	0,024	0,14
20	4A200L6Y3	30	2,1	90,5	0,90	3,7	0,046	0,12	0,022	0,13
21	4A225M6Y3	37	1,8	81,0	0,89	3,7	0,042	0,10	0,019	0,13
22	4A250S6Y3	45	1,4	90,5	0,89	3,8	0,037	0,090	0,015	0,14
23	4A250M6Y3	55	1,3	90,5	0,89	3,4	0,034	0,083	0,014	0,13
24	4A280S6Y3	75	2,0	92,0	0,89	3,7	0,032	0,122	0,021	0,13
25	4A280M6Y3	90	1,8	92,5	0,89	3,5	0,030	0,11	0,019	0,12
26	4A315M6Y3	132	1,7	93,5	0,90	3,5	0,023	0,090	0,018	0,12
27	4A355S6Y3	160	1,4	93,5	0,90	3,8	0,020	0,10	0,015	0,14
Синхронна частота обертання 750 об/хв										
28	4A132M8Y3	5,5	4,1	83,0	0,74	1,8	0,070	0,11	0,061	0,19
29	4A160M8Y3	11	2,5	87,0	0,75	2,0	0,066	0,13	0,031	0,18
30	4A180M8Y3	15	2,6	87,0	0,82	2,4	0,064	0,13	0,030	0,17
31	4A200M8Y3	18,5	2,3	88,5	0,84	2,6	0,057	0,13	0,026	0,16
32	4A200L8Y3	22	2,7	88,5	0,84	3,1	0,062	0,14	0,029	0,18
33	4A225M8Y3	30	1,8	90,5	0,81	2,3	0,045	0,12	0,022	0,17
34	4A250S8Y3	37	1,6	90,0	0,83	2,8	0,047	0,11	0,018	0,18
35	4A250M8Y3	45	1,4	91,0	0,84	2,6	0,037	0,099	0,016	0,17

Б2 – Технічні параметри двигунів постійного струму

№ з/п	Потужність P_{2H} , кВт	Напруга U_n , В	Частота обертання n_n , об/хв	ККД η_n , %	Параметри якоря та обмотки збудження		
					R_a , Ом	R_{Oa} , Ом	L_a , мГн
1	0,17	110	3000	47,5	5,84	610	128
2		220	1500	48,5	27,2	162	514
3	0,25	110	3000	61,5	2,52	610	78,7
4		220	2000	57	15,5	612	297
5	0,37	110	3000	61,5	2,52	610	48
6		220	2250	61,5	10,6	612	190
7	0,5	110	4000	65	1,79	470	53
8		220	2000	66	7,05	120	222
9	0,75	110	4000	71	0,81	359	26
10		220	4300	71,5	3,4	103	104
11	1	110	4000	71,5	0,6	365	12
12		220	4000	72,5	2,52	92	48
13	1,2	110	4000	75	0,44	259	14
14		220	4000	76,5	1,79	103	53
15	1,5	110	4000	70	0,42	181	4,8
16		220	4000	70	1,77	44	19,5
17	2	110	4000	78,5	0,2	265	6,6
18		220	4000	79	0,81	73	26
19	2,5	110	4000	76	0,196	156	2,3
20		220	4000	76	0,788	39,4	9,3
21	3	110	3000	75,5	0,138	128	18,5
22		220	2500	76,5	0,752	34,3	24,8
23	4,5	110	4000	78,5	0,11	108	6,4
24		220	3000	79,5	0,41	26,8	15,9
25	7	110	4000	81	0,067	111	18,7
26		220	4000	83	0,226	25,6	156
27	10,5	220	4000	84	0,14	111	98
28		440	4000	85	0,564	25,6	9,2
29	13	220	4000	85,5	0,081	61,5	14,4
30		440	4000	86,5	0,279	16,4	7,5
31	18	220	4000	87	0,037	53,1	123
32		440	4000	87,5	0,0145	12,6	34,2

Навчальне видання

ВОЛЯНСЬКИЙ
Сергій Михайлович,
канд. техн. наук

ВОЛЯНСЬКА
Яна Богданівна
доцент, канд. техн. наук

МОДЕЛЮВАННЯ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ

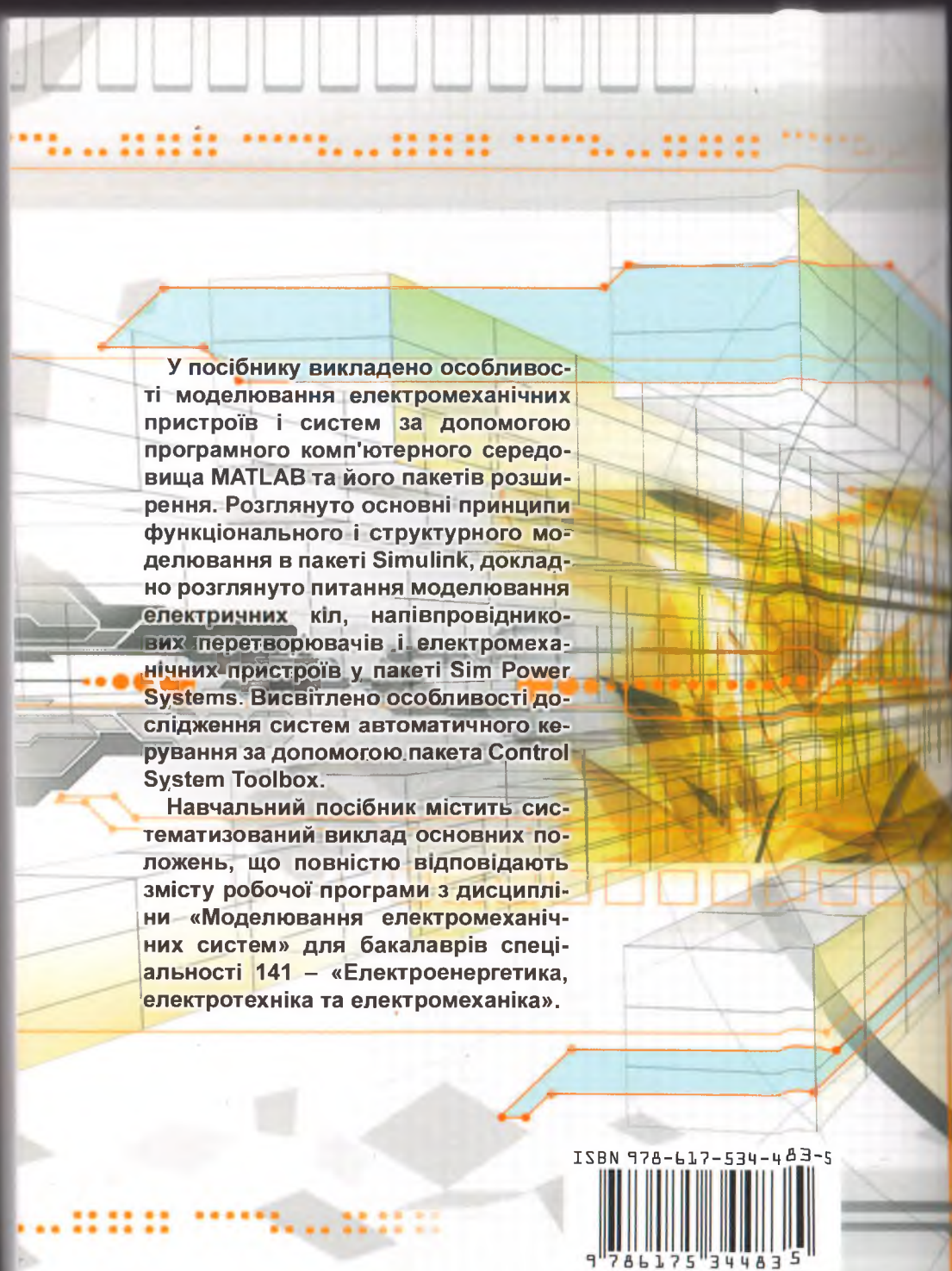
*Навчальний посібник
за редакцією доктора технічних наук,
професора О. А. Онищенка*

Формат 60x84/16. Ум. друк. арк. 14,3. Тираж 300 пр. Зам. № 534-483.

Видавець і виготовлювач
Товариство з обмеженою відповідальністю фірма «Іліон».
54038, м. Миколаїв, вул. Бузника, 5/1.
Свідоцтво суб'єкта видавничої справи
ДК № 1506 від 25.09.2003 р.

ДЛЯ НОТАТОК

THE
JOURNAL
OF
THE
ROYAL ANTHROPOLOGICAL INSTITUTE
VOLUME 100
PART 1
2000



У посібнику викладено особливості моделювання електромеханічних пристроїв і систем за допомогою програмного комп'ютерного середовища MATLAB та його пакетів розширення. Розглянуто основні принципи функціонального і структурного моделювання в пакеті Simulink, докладно розглянуто питання моделювання електричних кіл, напівпровідникових перетворювачів і електромеханічних пристроїв у пакеті Sim Power Systems. Висвітлено особливості дослідження систем автоматичного керування за допомогою пакета Control System Toolbox.

Навчальний посібник містить систематизований виклад основних положень, що повністю відповідають змісту робочої програми з дисципліни «Моделювання електромеханічних систем» для бакалаврів спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка».

ISBN 978-617-534-483-5



9 786175 344835